Universidad Carlos III de Madrid

INGENIERÍA SUPERIOR INFORMÁTICA

PROYECTO FINAL DE CARRERA

INSTALACIÓN Y CONFIGURACIÓN DE UN SERVIDOR GROUPWARE

AUTOR: SERGIO MONTOIRO PEINADO

TUTOR: ALEJANDRO CALDERÓN MARCOS

Junio 2013

Por todos estos momentos
y todos aquellos que vamos a compartir.
Gracias Ana.

Don’t worry head.
The computer will do all the thinking from now on.
Homer Simpson.

Agradecimientos

Creo que esta es una de las partes más difíciles pero más gratificantes de cualquier proyecto, el poder dar las gracias a todos aquellos que te han ayudado a llevarlo a cabo, esa es la parte gratificante, lo difícil, no dejarte alguien sin recordar.

En primer lugar agradecer a mis padres la oportunidad que me brindaron al volcar sobre mí todos los esfuerzos posibles, que no fueron pocos, para que pudiese seguir adelante con mis estudios. Sin ellos nada de esto habría comenzado, y por lo tanto no podría haber finalizado. A Ana, mi novia y futura esposa, la que día tras día, noche tras noche, está detrás de mi insistiendo en que acabe este proyecto y ponga un broche de oro a mi carrera universitaria, gracias Ana. Este proyecto es también parte tuya, te aseguro que algún día tendrás que leértelo.

Por otra parte recordar al resto de la familia, a mis hermanos, que también han estado detrás mía durante estos últimos años insistiendo en que acabase el dichoso proyecto. Que me han ayudado mucho en los momentos difíciles y que se que siempre podré contar con ellos para lo que haga falta. Mención especial para la recién incorporada a la familia, mi sobrina Alma, que aunque no sepa de que va esto, ya forma parte de nosotros. Acordarme de mi abuela, Santa, que nunca pudo ver a su nieto con la carrera acabada, pero que seguro que allá donde esté ahora puede sentirse orgullosa de que tiene un nieto Ingeniero.

Recordar a los buenos amigos, Alberto y Ana, que una vez fueron compañeros de universidad, y ahora han pasado a ser una de las parejas más cercanas, que en todo momento han ofrecido su apoyo y comprensión, y con los que he pasado muy buenos momentos. Recordar también a toda una vida de amistad con mis compañeros del Seminario menor de Rozas de Puerto Real, Baltasar, Javi, Simón (junto a Yoli y su hija Sofía), Juanjo, y muchos más, de los que tengo unos gratos recuerdos adolescentes que han servido para forjar unos lazos que perduran en el tiempo. Un rinconcito tam-

bién se merece mi nueva familia, la familia de Ana, que de un tiempo a esta parte me está tratando como uno más, y con los que me siento muy bien.

Como no acordarse del entorno universitario, de toda una época de descubrimiento y realización personal. Recordar a los amigos allí conocidos como Rubén, José Luis, Pablo, Toni, Roberto, Bris, etc. con los que he pasado muy buenos momentos, y con los que he compartido prácticas, equipos de fútbol, cañas, bocadillos y un sin fin de experiencias muy gratificantes. Mención especial al Laboratorio de Informática de la Universidad Carlos III, donde estuve como becario durante algo más de 2 años, y donde conocí a gente como Óscar y Roberto los técnicos con los que peleé mucho en su día para que todo funcionase perfectamente en los laboratorios, acordarme de José Luis, Yoshi, mi compañero de beca, una máquina en cuanto a Linux se refiere, y del que se me pegó la pasión por la administración y las cosas bien hechas. Y por supuesto acordarme de mi prima Laura, con la que conviví tanto tiempo durante la época universitaria, y tuvo que sufrir el compartir piso con un universitario como yo.

Dar las gracias a mi actual empresa Qwi-Tecnologías de la Información, por brindarme la oportunidad de trabajar con ellos, y con el equipo completo, Gustavo, Manu, Jesús, Pedro, y alguno más que hay por ahí.

Acordarme de mi tutor, Alejandro, ya que gracias a él esto se ha podido llevar a cabo, y esperemos que acabe en buen puerto.

¡Gracias a todos!

Resumen

El siguiente proyecto consiste en la instalación y configuración de un servidor groupware para un entorno empresarial. Dicho servidor estará compuesto de las herramientas necesarias para la comunicación entre todas las personas que conforman la empresa, así como la posibilidad de compartir cierta información con terceros.

Para un correcto funcionamiento de todas las partes, hay que dotar al servidor de una serie de servicios para su correcto funcionamiento:

- Sistema operativo.
- Servidor de correo.
- Servidor de directorio.
- Servidor de agenda.

Para que todo esto funcione a la perfección, el sistema operativo tendrá algunos servicios extra que conformen un ecosistema completo.

Palabras claves: Groupware, servidor de correo, servidor de directorio, servidor de agenda.

Abstract

This project involves the installation and configuration of a Groupware Server in a business environment. This server will be compounded by the necessary tools for communication between all bussiness environment.

For the best performance, the server must have all these services up and running:

- Operative System.
- Mail Server.
- Directory Server.
- Calendar Server.

All this pieces must work perfectly, for this reason, the operative system has some extra services that makes all a perfect ecosystem.

Keywords: Groupware, Mail Server, Directory Server, Calendar Server.

Índice general

Lista de Figuras

Lista de Tablas

Lista de Fragmentos de Código

1

Introducción

Este capítulo tiene el objetivo de realizar una breve introducción del proyecto, indicando el origen y fin del mismo, además de mostrar la estructura del documento.

1.1. Origen

Este proyecto ha sido realizado en un entorno empresarial para la gestión de los recursos internos de comunicación del personal mediante las herramientas de correo electrónico y agenda personal. Los nuevos tiempos y la necesidad de contar con las últimas tecnologías, hacían del antiguo servidor de correo un lastre a la hora de administrar y actualizar. Esto, unido a las necesidades de agrupar agendas y contactos en un mismo servidor, dota al proyecto de un mayor alcance. De esta forma se logra satisfacer las necesidades de sincronización entre distintos dispositivos, mantener la agenda siempre actualizada y no depender de servicios externos.

Los recursos hardware necesarios para un sistema de estas características son mínimos, permitiendo su ejecución en entornos poco potentes, ya que la mayoría de las operaciones realizadas son ejecutadas brevemente y consumiendo pocos recursos. Esto hace posible la instalación de dicho servidor en máquinas relativamente antiguas, máquinas con poca potencia o en entornos virtualizados. De esta forma se consigue ahorrar costes y mantener los datos físicos en dispositivos en posesión de la empresa. Este último punto es muy importante ya que al almacenar datos de usuario se deben cumplir las Leyes de Protección de Datos que rigen en el país donde la empresa está registrada.

1.2. Objetivo

El objetivo principal del proyecto consiste en el análisis, diseño e implementación de una solución Groupware para un entorno empresarial. El fin último del proyecto es ofrecer un sistema que supera en prestaciones al anterior, y ofrece fiabilidad, flexibilidad, disponibilidad y seguridad.

Para llevar a cabo el objetivo, se han utilizado las distintas máquinas del entorno de la empresa, además del software necesario en materia de los distintos sistemas desplegados para ofrecer los servicios finales.

El nuevo sistema debe ofrecer un mayor grado de comunicación entre el personal de la empresa, así como ofrecer mayor organización entre las distintas partes. De esta forma el servidor será capaz de mantener siempre actualizados y disponibles los servicios de:

- Correo electrónico.
- Directorio.
- Agenda.

Además de servicios secundarios como:

- Autenticación centralizada.
- Servidor de páginas web.
- Certification Authority (CA).

1.3. Medios empleados

Para la realización de este proyecto se han utilizado los siguientes medios:

- Equipo de desarrollo, portátil con la distribución GNU/Linux Linux Mint.
- Máquinas virtuales en servidor VMWare ESXi y Virtual Box.
- Máquinas virtuales con la distribución GNU/Linux Ubuntu Server.
- Memoria realizada en LaTeX.

1.4. Estructura del documento

El documento está dividido en los siguientes capítulos:

- **Introducción** donde se exponen el origen y los objetivos del proyecto, además de la estructura del presente documento.
- **Estado de la cuestión**, donde se expone toda la información relativa a los servicios que se pretenden instalar, así como las distintas soluciones que se pueden encontrar actualmente en el mercado.
- **Análisis**, donde se analizará el estado actual y la solución elegida junto a los motivos de su elección.
- **Diseño**, en este apartado se detallará el diseño final escogido para su implantación.
- **Presupuesto**, después del análisis y diseño se incluirá el presupuesto que se extrae de la información anterior.
- **Implantación**, donde se explican en detalle, los pasos llevados a cabo para la consecución del proyecto.
- **Resultados**, aquí se muestran los resultados finales del proyecto.
- **Conclusiones, opinión personal y trabajos futuros**, donde se detallan las conclusiones extraídas en base al funcionamiento de todas las partes del sistema. También se abordarán las cuestiones sobre los trabajos pendientes y tareas a realizar en el futuro.
- **Apéndices y bibliografía**.

2

Estado de la cuestión

Aquí se analizan las diferentes infraestructuras necesarias para la creación del sistema de este proyecto.

2.1. Sistemas operativos

Un sistema operativo (SO) es un conjunto de programas destinados a permitir la comunicación del usuario con un dispositivo electrónico, además de gestionar sus recursos de manera eficiente. El SO se pone a funcionar una vez que el usuario enciende el dispositivo, y gestiona los recursos hardware de la máquina desde los niveles más básicos.

Un SO puede encontrarse en la mayoría de los aparatos electrónicos que se utilizan a diario sin necesidad de estar conectados a un ordenador y que utilicen microprocesadores para funcionar, de esta forma el usuario puede utilizar los recursos físicos o hardware de la máquina para que ésta cumpla sus funciones.

Los distintos dispositivos electrónicos pueden abarcar aparatos del más amplio rango, desde televisores, teléfonos móviles, equipos de música, reproductores Blueray... y ordenadores. Existen distintas familias de SO que serán enumerados en las siguientes secciones.

2.1.1. Microsoft Windows

☞ http://windows.microsoft.com/

El sistema operativo Windows pertenece a la empresa de software Microsoft. Dicho sistema operativo comenzó a desarrollarse en la década de los 70 [5]. Tiene distintas versiones enfocadas a distintos tipos de dispositivo y usuario final. Se pueden diferenciar tres grandes grupos de versiones:

- **Enfocados a escritorio**, suelen diferenciarse del resto ya que poseen una interfaz gráfica para su uso. El usuario suele interactuar con el sistema con dispositivos como el ratón y el teclado. Su principal función es que el usuario estándar pueda ejecutar sus herramientas de ofimática, internet, etc.

- **Enfocados a dispositivos móviles**, la principal diferencia con el anterior es que el usuario posee un dispositivo con entrada táctil en pantalla para su uso. De esta forma el número de periféricos disminuye, con lo que aumenta su portabilidad.

- **Enfocados a servidores**, en este último grupo se encuentran los sistemas encargados de ofrecer servicios al usuario, la mayoría de las veces de forma remota. En las últimas versiones de Windows se permite realizar instalaciones *core* [9] lo que permite simplificar la carga de trabajo del sistema al no poseer interfaz gráfica completa.

Al realizar una instalación para un sistema de servicios, se analizará con detenimiento la última de las versiones nombrada. En la actualidad Windows posee distintas versiones de su sistema enfocado a servidores:

- Microsoft Windows Server 2000
- Microsoft Windows Server 2003
- Microsoft Windows Server 2008
- Microsoft Windows Server 2012

Cada una de las distintas versiones, y más a partir de la versión 2008, tiene a su vez distintos tipos de licencias [8] [10], que dotan al usuario final de una mayor flexibilidad a la hora de realizar la instalación del SO, aunque a veces también son un quebradero de cabeza para los administradores.

Microsoft Windows Server 2008

Los sistemas de Microsoft se caracterizan por tener un ciclo de vida bastante amplio [6], por lo tanto, y siguiendo la tabla 2.1, la versión 2008 de Windows Server es la más indicada para realizar una instalación, ya que es una versión estable, que lleva en el mercado más de 3 años y dispone de una amplia serie de parches que han solucionado distintos problemas de seguridad y/o compatibilidad.

Producto	Inicio ciclo de vida	Fin soporte técnico principal	Fin soporte técnico extendido
Windows Server 2012 Standard	30/10/2012	09/01/2018	10/01/2023
Windows Server 2008 R2 Enterprise	22/10/2009	13/01/2015	14/01/2020
Windows Server 2003, Enterprise Edition	28/05/2003	13/07/2010	14/07/2015

Tabla 2.1: Ciclo de vida de las principales versiones de Microsoft Windows Server

Microsoft Windows Server 2008 R2 es un SO creado por Microsoft que tuvo su salida en el mercado en el año 2008. Esta versión del SO para servidores, está basada en la versión 6.X del núcleo de Windows. Las mejoras que incluye este sistema respecto a sus versiones anteriores son nuevas funcionalidades para el Active Directory, nuevas prestaciones de virtualización y administración de sistemas, la inclusión de IIS 7.5 y el soporte para más de 256 procesadores.

De esta forma se convierte en uno de los candidatos idóneos a la hora de plantear un equipo para gestionar las tareas de un servidor. Las características de este SO junto a la cantidad de servicios que provee, dotan al servidor de un amplio conjunto de aplicaciones todas ellas apoyadas por la empresa Microsoft.

Las ventajas y desventajas que estas características proveen son las siguientes:

Ventajas

- Facilidad de uso al tener múltiples herramientas en entorno gráfico para su administración.

- Gestión user friendly que no necesita de manos expertas para realizar tareas básicas de mantenimiento.
- Soporte de pago. La empresa Microsoft es la encargada de mantener el sistema actualizado y libre de fallos o vulnerabilidades, proveyendo de parches correspondientes.
- Alta compatibilidad con el hardware del mercado, ya que muchos sistemas ejecutan Windows, y las empresas de hardware compilan drivers específicos para este SO

Desventajas

- El código fuente es cerrado, por lo que no se pueden agregar particularidades o gestionar a más bajo nivel el SO
- El coste es alto, tanto a nivel de software como de hardware. Los SO de Microsoft se caracterizan por un alto consumo de requisitos hardware, si a esto le sumamos el coste de las pertinentes licencias, el resultado es una suma de dinero importante.
- Todas las partes del sistema tienen una fuerte relación unas con otras, haciendo que el fallo de una de ellas pueda dejar el resto de servicios inutilizables.

2.1.2. GNU/Linux

☞ http://www.gnu.org/gnu/linux-and-gnu.html/

☞ http://www.ubuntu.com/

☞ http://www.debian.org/

☞ http://www.redhat.com/

☞ http://www.centos.org/

☞ https://www.suse.com/

Los sistemas GNU/Linux son aquellos SO que se componen de dos partes principales: el núcleo del sistema operativo llamado Linux, y las herramientas creadas por la fundación GNU. Es uno de los paradigmas del desarrollo de software libre (y de código abierto), donde el código fuente está disponible públicamente y cualquier persona, con los conocimientos informáticos adecuados, puede libremente estudiarlo, usarlo, modificarlo y redistribuirlo.

Dicho SO se compone, además de las herramientas anteriormente descritas, de multitud de software basado en la filosofía del software libre. La unión de estas piezas componen un conjunto de servicios y programas, que se denominan como distribuciones GNU/Linux.

Las distribuciones GNU/Linux son colecciones de software que suelen contener grandes cantidades de paquetes, además del núcleo. El software que suelen incluir consta de una enorme variedad de aplicaciones como: entornos gráficos, suites ofimáticas, servidores web, herramientas de desarrollo, compiladores, etc. De forma coloquial se aplica el término Linux a estas distribuciones, aunque esto sea incorrecto. Esto es así ya que la forma más sencilla, simple y popular de conseguir un sistema GNU/Linux es realizando la instalación de una de estas distribuciones.

La amplia mayoría de distribuciones GNU/Linux tienen ese carácter de software libre y gratuito, pero debido a la gran cantidad de las mismas, se pueden encontrar en el mercado distribuciones comerciales, que en la mayoría de los casos ofrecen al usuario final un amplio abanico de posibilidades en cuanto al soporte y fiabilidad de los sistemas. Es por ello que en la tabla 2.2 se especifican las versiones más populares a nivel de servidor, indicando en cada caso las últimas versiones liberadas, así como las características más relevantes de las distintas distribuciones [21].

Producto	Última versión y fecha	Características
Debian	6.0.6 Squeeze (29/09/2012)	Basado en el gestor de paquetes apt. Desarrollado totalmente por la comunidad. Suele tener versiones antiguas pero muy estables de los paquetes utilizados.
Ubuntu	12.10 Quantal Quetzal (18/10/2012)	Al estar basado en Debian, también esta basado en el gestor de paquetes apt. Desarrollado por la empresa Canonical, ofrece un ciclo de vida amplio y versiones cada 6 meses.

Continúa ...

Producto	Última versión y fecha	Características
RedHat	6.3 RHEL6 (20/06/2012)	Basado en el gestor de paquetes yum, que utiliza rpm. Esta distribución requiere de un pago por utilizarla.
Centos	6.3 (09/07/2012)	Este sistema operativo esta basado totalmente en RedHat. Ofrece una alternativa libre a las versiones liberadas del anterior.
Suse	11 SP2 (15/02/2012)	Sistema basado en paquetes rpm y gestionado por yast2. Al igual que RedHat también requiere de pago por utilización. Existe una alternativa denominada OpenSuse que es mantenida por la comunidad.

Tabla 2.2: Principales distribuciones GNU/Linux enfocadas a servidores

Ventajas

- Ofrece funcionalidades de servidor sin necesidad de poseer un interfaz gráfico, haciendo el uso de recursos más eficiente.
- Sus costes en cuanto a software suelen ser cero, y el soporte por múltiples arquitecturas hace que los costes de hardware puedan ser también bajos.
- La mayoría de las distribuciones y paquetes son sostenidos por grandes fundaciones o grupos de usuarios muy activos, proveyendo actualizaciones y parches de seguridad muy a menudo.
- Tanto el sector privado como el público está volcándose con este tipo de aplicaciones que evita el desembolso de cantidades de dinero por licencias.

Desventajas

- Requiere de mayores conocimientos técnicos para su administración.

- El soporte del software puede depender de la comunidad de usuarios, a los que no se les puede exigir ni tiempos ni funcionalidades extra.

2.1.3. Otros SO

Dentro de esta categoría se engloban distintos sistemas operativos que no están dirigidos a realizar tareas de servidor, o pertenecen a otra familia distintas de los sistemas de Windows y GNU/Linux.

Unix

☞ http://www.unix.org/

Como uno de los SO más importantes en la actualidad, el sistema UNIX es sin duda una interesante aplicación que cuenta con diversas utilidades y funciones. Creado a fines de los años sesenta para la empresa AT&T y GE, este SO es forma parte de un importante grupo de familias fácilmente reconocibles por el usuario común de las tecnologías de la información.

Unix es un SO multiuso que puede ser portable. En sus inicios, este sistema no encontró muy buenos resultados por tratarse de un sistema lento y de relativa eficiencia. Sin embargo, con la perseverancia de Ken Thompson, uno de sus creadores, el sistema Unix finalmente vio la luz al recibirse el apoyo económico de las empresas que lo habían iniciado. Luego de su creación ganó popularidad masiva y hasta llegaron a desarrollarse productos similares pero no oficiales con tal de expandir su uso en el público general.

El sistema Unix está íntimamente relacionado con la aparición de internet ya que fue el que instaló la idea de cliente y servidor así como también al generar la disposición en red de las computadoras utilizadas en vez de funcionar de manera individual en computadoras aisladas. Su utilidad primordial es la de almacenamiento de información a través de un sistema de archivos jerarquizado. También cuenta con numerosas herramientas de software y con la posibilidad de ser utilizado de igual manera y al mismo tiempo por varios usuarios.

Se habla de Unix como un conjunto de familias entre las cuales encontramos AT&T, Xenix, Linux y muchas otras. Todas ellas han estado relacionadas con el sistema operativo en cuanto lo han licenciado y utilizado. Actualmente Unix forma parte de *The Open Group* y su utilización es

exclusivamente dependiente de la autorización *Single Unix Specification* con el fin de evitar su uso ilegal por parte de otras compañías.

Entre las implementaciones más importantes de este sistema operativo cabe destacar:

- **Solaris**, desarrollado por Sun Microsystems, recientemente adquirida por Oracle Inc. Conocido por su gran estabilidad, Solaris, es uno de los SO más difundidos a nivel empresarial.
- **HP/UX**, desarrollado por HP, cuyo desarrollo está ligado a los servidores del fabricante.
- **AIX**, desarrollado por IBM, y al igual que los anteriores su código es propietario.
- **FreeBSD**, primera solución de software libre Unix, considerada muy fiable y robusta.
- **OpenSolaris**, solución que ganó popularidad tras la adquisición de Sun Microsystems por parte de Oracle Inc. Es la rama de código abierto de Solaris.

Mac OS

☞ http://www.apple.com/osx/

Mac OS X es una línea de SO gráfico desarrollado y distribuido por la compañía Apple Inc, especialmente para ser usados en computadoras Macintosh y/o dispositivos como el iPhone, el iPod y similares.

Mac OS X es el sucesor del original Mac OS de 1984, primer SO de Apple. Pero, a diferencia de su predecesor, el Mac OS X está basado en los SO Unix. Fue en el año 1997 cuando Steve Jobs, nombrado CEO de Apple, decidió terminar con la versión clásica y crear este nuevo SO usando tecnología del SO NEXTSTEP de la compañía NeXT (adquirida por Apple a principios de ese año).

La primera versión fue lanzada en 1999 con el nombre de Mac OS X Server 1.0, seguida por una versión orientada a escritorio, la Mac OS X v10.0 en marzo de 2001.

Las versiones para dispositivos pequeños, como los iPhone y los iPod, son versiones reducidas del SO, denominadas iOS y que se detallarán en la sección Sistemas operativos para móviles, que se encuentra en la página 41.

En la siguiente tabla 2.3 se detallan las últimas versiones de este SO

Producto	Última versión y fecha
Max OS X Server 1.0	Hera 1.2v3 27/10/2000
10.0	Cheetah 10.0.4 22/06/2001
10.1	Puma 10.1.5 06/10/2002
10.2	Jaguar 10.2.8 03/10/2003
10.3	Panther 10.3.9 15/04/2005
10.4	Tiger 10.4.11 14/11/2007
10.5	Leopard 10.5.8 23/06/2009
10.6	Snow Leopard 10.6.8 23/06/2011
10.7	Lion 10.7.4 21/05/2012
10.8	Mountain Lion 10.8.2 04/10/2012

Tabla 2.3: Versiones de Mac Os X [23]

Sistemas operativos para móviles

Un SO móvil es un sistema que controla un dispositivo móvil. Sin embargo, los SO móviles son mucho más simples y están más orientados a la conectividad inalámbrica, los formatos multimedia para móviles y las diferentes maneras de introducir información en ellos.

- **Android**, Android es un SO orientado a dispositivos móviles, basado en una versión modificada del núcleo Linux. Inicialmente fue desarrollado por Android Inc., una pequeña empresa, que posteriormente fue comprada por Google; en la actualidad lo desarrollan los miembros de la Open Handset Alliance (liderada por Google).

- **iOS**, es un sistema operativo desarrollado por Apple originalmente para su teléfono iPhone, pero lo emplean otros de los productos de la compañía como el iPod Touch, iPad y Apple TV.

- **BlackBerry OS**, es un sistema operativo para móviles desarrollado por Research In Motion (RIC) para su línea de smartphone BlackBerry. La plataforma BlackBerry es muy conocida

por su soporte nativo al correo electrónico corporativo a través de MIDP, que permite activación inalámbrica completa y sincronización con Microsoft Exchange, Lotus Domain, o Novell GroupWise.

- **Symbian OS**, fue producto de la alianza de varias empresas de telefonía móvil, dentro de las que se encuentran Nokia, Sony Ericsson, Samsung y Siemens. Técnicamente, el sistema operativo Symbian es una colección compacta de código ejecutable y varios archivos, la mayoría de ellos son bibliotecas vinculadas dinámicamente (DLL por sus siglas en inglés) y otros datos requeridos, incluyendo archivos de configuración, de imágenes y de tipografía, entre otros recursos residentes. Symbian se almacena, generalmente, en un circuito flash dentro del dispositivo móvil. Gracias a este tipo de tecnología, se puede conservar información aun si el sistema no posee carga eléctrica en la batería, además de que le es factible reprogramarse, sin necesidad de separarla de los demás circuitos.

- **Windows Phone**, es un sistema operativo móvil desarrollado por la empresa Microsoft para teléfonos inteligentes y otros dispositivos móviles. Fue lanzado al mercado el 21 de octubre de 2010 en Europa y el 8 de Noviembre en Estados Unidos, con la finalidad de suplantar el conocido Windows Mobile.

Dado su escaso consumo de recursos y al estar basados en otros sistemas orientados a PC's, se están llevando a cabo estudios e investigaciones para utilizar dichos sistemas como máquinas orientadas a servicios, sobre todo con la plataforma Android [1] [15].

2.2. Servidores de correo

Un servidor de correo es una aplicación que nos permite enviar mensajes (correos) de unos usuarios a otros, con independencia de la red que dichos usuarios estén utilizando.

Para lograrlo se ha especificado una serie de protocolos, cada uno con una finalidad concreta:

- Simple Mail Transfer Protocol (SMTP), se basa en una entrega punto a punto, un cliente SMTP contacta con el servidor SMTP del host destino para entregarle directamente el correo, este nos da seguridad en la entrega al receptor ya que espera que sea guardado con éxito.

- Post Office Protocol (POP), le permite a los clientes obtener los mensajes que se encuentran almacenados en el servidor. El protocolo POP después de descargar el mensaje lo guarda en el disco y puedo abrirlo sin necesidad de estar conectado.

- Internet Message Access Protocol (IMAP), es un protocolo de red de acceso a mensajes electrónicos almacenados en un servidor. Mediante IMAP se puede tener acceso al correo electrónico desde cualquier equipo que tenga una conexión a Internet.

Para que un servidor de correo funcione correctamente, al menos debe cumplir el protocolo SMTP para el envío de correos, y uno o los dos protocolos de recepción/visualización, como son POP e IMAP.

Para realizar el envío de un correo electrónico entran en juego una serie de programas o agentes, encargados de realizar el intercambio de información siguiendo los protocolos específicos. Por ello se especifican los siguientes agentes:

- Mail User Agent (MUA), sistema utilizado por el usuario para el envío y recepción de correos electrónicos.

- Mail Delivery Agent (MDA), sistema encargado de almacenar y repartir los mensajes en los buzones de los destinatarios.

- Mail Transport Agent (MTA), sistema encargado de enviar mensajes entre distintos servidores de correo.

El envío y recepción de un correo se resume en los siguientes pasos:

1. El usuario envía un correo electrónico a través de su programa MUA

2. El servidor MTA recibe el correo y comprueba la dirección del mismo. Realiza las siguientes tareas según la dirección:

   a) Si la dirección es externa, se lo envía al siguiente MTA y vuelve a 2

   b) Si la dirección es interna, se lo envía al MDA para que lo almacene

3. Por último el usuario receptor recibirá una notificación en su MUA para poder descargarse el mensaje recibido.

En las siguientes subsecciones se detallarán algunas herramientas utilizadas como servidores de correo.

2.2.1. Cyrus IMAP

☞ http://cyrusimap.web.cmu.edu/

Cyrus IMAP es un MDA desarrollado y mantenido por el Andrew Systems Group de la Carnegie Mellon University [16].

A diferencia de otros servidores IMAP, Cyrus usa su propio método para almacenar el correo de los usuarios. Cada mensaje es almacenado en su propio fichero. El beneficio de usar ficheros separados es una mayor fiabilidad ya que sólo un mensaje se pierde en caso de error del sistema de ficheros. Los metadatos, tales como el estado de un mensaje (leído, etc.) se almacenan en una base de datos. Además, los mensajes son indexados para mejorar el rendimiento de Cyrus, especialmente con muchos usuarios e ingentes cantidades de mensajes.

Otra característica muy importante es que no son necesarias cuentas locales de GNU/Linux para cada usuario. Todos los usuarios son identificados por el servidor IMAP. Esto lo convierte en una magnífica solución cuando se tiene una gran cantidad de usuarios.

La administración es llevada a cabo mediante comandos especiales de IMAP. Esto le permite usar tanto la interfaz de línea de comandos como los interfaces web. Este método es mucho más seguro que un interfaz web para */etc/passwd*.

Desde la versión 2.1 de Cyrus, se usa la versión 2 de la librería SASL para la identificación.

Ventajas

- La identificación es independiente del sistema operativo, por lo que no es necesario mantener los usuarios en el SO.
- La indexación y guardado de los mensajes en formato maildir.

Desventajas

- No es multiplataforma, sólo puede utilizarse en máquinas GNU/Linux.
- Es necesario realizar instalaciones de otras herramientas para tener una interfaz amigable de administración.
- Como MDA no soporta el protocolo SMTP por lo que debe ser utilizado en conjunto a otra herramienta, Postfix, para tener un servidor de correo completo.

2.2.2. Dovecot

☞ http://www.dovecot.org

Dovecot es un servidor de POP3 e IMAP, de código fuente abierto, que funciona en GNU/Linux y sistemas basados sobre Unix y diseñado con la seguridad como principal objetivo. Dovecot puede utilizar tanto el formato mbox, maildir y dbox, y es compatible con las implementaciones de los servidores UW-IMAP y Courier IMAP.

Las características principales de Dovecot es la capacidad de comunicación con multitud de clientes además de con herramientas propietarias, como pueden ser la de Microsoft. Además está construido para optimizar al máximo los recursos de las máquina proveyendo de índices y gestionando los errores de forma que el usuario siempre tiene lo que necesita utilizando para ello el menor tiempo posible. Además la administración se realiza desde interfaces user friendly que facilitan la tarea de los administradores menos expertos.

Ventajas

- Posee herramientas de administración user friendly.
- Integra multitud de sistemas de identificación [17].
- Es compatible con la mayoría de los clientes de correo, implementando multitud de protocolos de acceso.
- Permite la integración de plugins que extienden su funcionalidad.

Desventajas

- No es multiplataforma, sólo puede utilizarse en máquinas GNU/Linux.
- Como MDA no soporta el protocolo SMTP por lo que debe ser utilizado en conjunto a otra herramienta, Postfix, para tener un servidor de correo completo.

2.2.3. Atmail

☞ http://atmail.com/

Es una plataforma comercial de mensajería creada bajo GNU/Linux por la empresa del mismo nombre. La empresa ofrece soluciones de correo electrónico web, servidores de correo electrónico y soluciones groupware construidas en entornos GNU/Linux pero siempre bajo licencias comerciales. Las últimas versiones de Atmail, ofrecen una amplia variedad de servicios a parte de los estrictamente asociados al correo electrónico y la comunicación entre las personas de la empresa. Dicho compendio de servicios incluyen almacenamiento en la nube, cliente de agenda, etc. Además permite la identificación de los servicios a través de distintos métodos como puede ser base de datos, ldap, archivos de texto, etc. También contiene software encargado de la seguridad del entorno y los mensajes que llegan al servidor, contando con antivirus y antispam.

Ventajas

- Entorno totalmente configurado y listo para usar. Utiliza una administración sencilla y es compatible con multitud de sistemas y protocolos.
- Construido con programas de código abierto.
- Es compatible con una amplia variedad de dispositivos, además implementa la mayoría de los protocolos utilizados por los clientes de correo.

Desventajas

- El precio por su licencia. Es software bajo licencias comerciales.

- Sólo puede utilizarse con la solución GNU/Linux planteada por el desarrollador, no permitiendo su instalación en otros sistemas.
- No permite la instalación de elementos externos y que puedan ser necesarios para la utilización dentro del entorno empresarial.

2.2.4. Microsoft Exchange

☞ http://www.microsoft.com/exchange/

Microsoft Exchange Server es un producto de software de colaboración entre usuarios desarrollado por Microsoft. Es parte de la línea de productos para servidores Microsoft Servers y está extendido su uso por grandes empresas.

Microsoft Exchange Server permite manipular correos electrónicos, almacenamiento de información, calendarios, contactos y tareas compartidas, etc. Posee funcionalidades destinadas a simplificar la administración así como facilita la movilidad de los usuarios que lo utilizan [7].

Además, Microsoft en esta última versión ha incluido entre sus características el empleo de métodos más eficaces a la hora de conservar todos los registros de los mensajes de correo, de modo que ante posibles acciones que requieran el uso de dichos correos ante las autoridades sea más fácil disponer de todos los datos disponibles de una forma más eficaz [7].

Ventajas

- Su compatibilidad con la mayoría de dispositivos está asegurada ya que es utilizado por la gran mayoría de las grandes empresas.
- Ofrece una interfaz de administración fácil para el usuario normal.
- Tiene servicios en la nube, que permiten el acceso a los archivos desde cualquier lugar, y a cualquier hora sin necesidad de mantener el propio hardware.

Desventajas

- Su precio de licencia es elevado.

- Requiere su utilización con un servidor Microsoft Windows, lo cual encarece más su licencia.

2.2.5. Postfix

☞ http://www.postfix.org/

Postfix es un servidor de correo de software libre para el enrutamiento y envío de correo electrónico, creado con la intención de que sea una alternativa más rápida, fácil de administrar y segura a Sendmail. A día de hoy postfix continúa siendo desarrollado activamente, su última versión data de Febrero del 2013, la versión 2.10. Postfix es el MTA por omisión en diversas distribuciones GNU/-Linux y en las últimas versiones de Mac OS. Dada su extensión en múltiples plataformas, existen infinidad de configuraciones y extensiones, que hacen de este software uno de los más extendidos en cuanto a MTA se refiere.

Ventajas

- Existen multitud de herramientas que añaden y amplían su funcionalidad.
- Es software libre y compatible con distintos sistemas operativos.
- Se generan versiones actualizadas cada poco tiempo, e incluso se corrigen vulnerabilidades de las últimas 4 versiones.

Desventajas

- Es difícil de administrar, y requiere de otras herramientas para tener una funcionalidad más completa.
- No soporta los protocolos POP e IMAP, por lo que se tiene que valer de otro software para ser un servidor de correo completo

2.2.6. Horde IMP

☞ http://www.horde.org/

Horde IMP es una aplicación webmail basada en el framework Horde que permite el acceso a buzones POP y IMAP.

El Framework Horde es un framework libre escrito en PHP, para el desarrollo de aplicaciones groupware basadas en la Web.

El Proyecto Horde se compone de unas bibliotecas (el mencionado Horde Framework) que proporcionan funcionalidades básicas (identificación, gestión de preferencias, interfaz gráfica, etc) y que funciona como nexo de unión entre distintas aplicaciones de usuario, que son gestionadas como sub-proyectos independientes.

El objetivo del proyecto es crear aplicaciones sólidas, basadas en estándares, multiplataforma y de fácil acceso para cualquier usuario, independientemente de su idioma o localización.

Además dentro del conjunto de proyectos de código libre de Horde se pueden encontrar:

- **Horde**: Se trata del framework sobre el que trabajan el resto de aplicaciones. Es el único módulo no opcional del sistema Horde.
- **IMP**: Sistema webmail que permite el acceso a buzones POP3 o IMAP.
- **MIMP**: Derivativo de IMP, con una interfaz mínima para hacerlo usable en dispositivos móviles.
- **DIMP**: Derivativo de IMP, con interfaz basado en AJAX, con el objeto de hacerlo más dinámico y rápido.
- **Ingo**: Sistema de gestión y aplicación de reglas de filtrado de correo.
- **Sork**: Conjunto de utilidades para el sistema webmail que permiten al usuario funciones como el cambio de contraseña, redirecciones, respuestas automáticas, etc.
- **Turba**: Agenda de contactos.
- **Mnemo**: Gestor de notas.
- **Kronolith**: Gestión de agendas y calendarios con funciones de grupo.
- **Gollem**: Gestor de archivos con posibilidad de usar un SGBD como backend.

- **Nag**: Gestor de listas de tareas.
- **Trean**: Gestor de favoritos.
- **Chora**: Interfaz al sistema CVS de código fuente del proyecto.
- **Whups**: Sistema de gestión de incidentes basado en boletines.

Además está escrito siguiendo los estándares web actuales, como son XHTML, MIME, ANSI SQL, etc. y es compatible con la mayoría de los protocolos de conexión para herramientas de este tipo, como puede ser iCalendar y vCard, syncML o WebDAV. Además es totalmente independiente de la tecnología utilizada como almacenamiento de datos, y están disponible los drivers y configuraciones para distintas bases de datos, así como conexión con sistemas LDAP.

Ventajas

- Construido con programas de código abierto. Y utilizable en cualquier sistema, ya que está basado en PHP, y soporta múltiples backend.
- Es compatible con una amplia variedad de dispositivos y protocolos.
- Ofrece multitud de herramientas y módulos listos para utilizar. La configuración de los mismos se realiza de forma sencilla a través de pantallas de un explorador web.
- Los usuarios disponen de multitud de herramientas todas ellas conectadas entre sí y accesibles desde un sólo sitio web.

Desventajas

- En ocasiones, y dada su facilidad de uso a través de la interfaz web, es difícil encontrar las configuraciones en los archivos físicos.

2.2.7. Zimbra

☞ http://www.zimbra.com/

El grupo de aplicaciones Zimbra contiene, entre otro software, un servidor de correo electrónico, así como un cliente de correo web. El software de Zimbra se distribuye de dos formas, una totalmente de código abierto y sin cargos, y otra que incluye componentes comerciales, por los que hay que adquirir una licencia para su uso. Además esta última opción da derecho a soporte por parte de la empresa que desarrolla la suite, en este caso VMWare. Todo este conjunto de herramientas ofrece un completo servidor de correo tanto para la recepción como el envío, así como ofrece los medios adecuados para su administración a partir de interfaces fáciles de utilizar por personas no expertas. Su integración con otros sistemas está un poco limitada, ya que el paquete software contiene todas las herramientas ya instaladas y listas para usar dentro del mismo servidor, lo que hace que sea difícil la configuración con otras plataformas externas.

Ventajas

- Entorno totalmente configurado y listo para usar. Utiliza una administración sencilla.
- Construido con programas de código abierto. Aunque el código final no es del todo distribuido.
- Es compatible con una amplia variedad de dispositivos.
- Ofrece sus propias herramientas de acceso al servidor ofreciendo una experiencia total al usuario final sin necesidad de elementos de terceros.

Desventajas

- El precio por su licencia. Aunque dispone de versión libre de pago, ésta no tiene las últimas actualizaciones ni dispone de toda la funcionalidad.
- Sólo puede utilizarse con la solución GNU/Linux planteada por el desarrollador, no permitiendo su instalación en otros sistemas.
- No permite la instalación de elementos externos y que puedan ser necesarios para la utilización dentro del entorno empresarial.

2.3. Servidores de directorio

Un sistema de estas características, necesita un proponer a sus usuarios una forma inequívoca de identificación. Cada persona conocerá su nombre de usuario y contraseña que introducirá cuando necesite acceso al sistema. Dichos datos serán contrastados contra una base de datos, que devolverá el resultado de comparar el nombre de usuario y contraseña contra los valores almacenados. Además de dicha información este sistema dispondrá de más información referente al usuario, dicha información comprende datos personales, dirección de correo, claves públicas de cifrado, etc. Existen multitud de sistemas y protocolos que cumplen estas características, a continuación se exponen las más utilizadas.

2.3.1. Network Information Service

☞ http://www.linux-nis.org/

Network Information Service (NIS), es el nombre del protocolo utilizado para el envío de datos de configuración entre sistemas distribuidos. Entre los datos distribuidos se pueden encontrar nombres de usuarios y máquinas que componen una red.

NIS proporciona prestaciones de acceso a bases de datos genéricas que pueden utilizarse para distribuir, por ejemplo, la información contenida en los ficheros *passwd* y *groups* a todos los nodos de su red. Esto hace que la red parezca un sistema individual, con las mismas cuentas en todos los nodos. De manera similar, se puede usar NIS para distribuir la información de nombres de nodo contenida en */etc/hosts* a todas las máquinas de la red.

A medida que las redes fueron evolucionando, NIS fue desapareciendo dejando paso a otros servidores de directorio como puede ser Lightweight Directory Access Protocol (LDAP).

Ventajas

- Utiliza un sistema de directorios, obteniendo un buen rendimiento en las búsquedas, que son la mayoría de las operaciones realizadas.

Desventajas

- Su compatibilidad con sistemas no UNIX es difícil de encontrar, y las soluciones que existen son de pago.

- Los datos enviados por la red no son cifrados, lo que hace que la distribución requiera de una mayor configuración para conservar la integridad y seguridad de los mismos.

2.3.2. OpenLdap

☞ http://www.openldap.org/

Es la implementación abierta del protocolo LDAP.

LDAP es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red. LDAP también se considera una base de datos (aunque su sistema de almacenamiento puede ser diferente) a la que pueden realizarse consultas. Un directorio es un conjunto de objetos con atributos organizados en una manera lógica y jerárquica. Un árbol de directorio LDAP refleja varios límites políticos, geográficos u organizacionales, dependiendo del modelo elegido. Los despliegues actuales de LDAP tienden a usar nombres de Domain Name System (DNS) para estructurar los niveles más altos de la jerarquía. Conforme se desciende en el directorio pueden aparecer entradas que representan personas, unidades organizacionales, impresoras, documentos, grupos de personas o cualquier cosa que representa una entrada dada en el árbol (o múltiples entradas). Habitualmente, almacena la información de autenticación (usuario y contraseña) y es utilizado para identificarse, aunque es posible almacenar otra información (datos de contacto del usuario, ubicación de diversos recursos de la red, permisos, certificados, etc). A manera de síntesis, LDAP es un protocolo de acceso unificado a un conjunto de información sobre una red. La versión actual es LDAPv3, que está especificada en una serie de Internet Engineering Task Force (IETF) Standard Track Request for Comments (RFCs) como se detalla en el documento RFC 4510 [20].

Las ventajas de utilizar el protocolo LDAP en los directorios se pueden enumerar como las siguientes:

- Al estar basado en un sistema de directorios, es muy rápido en la lectura de registros.
- Permite replicar el servidor de forma muy sencilla y económica.
- Muchas aplicaciones de todo tipo tienen interfaces de conexión a LDAP y se pueden integrar fácilmente.
- Dispone de un modelo de nombres globales que asegura que todas las entradas son únicas.
- Usa un sistema jerárquico de almacenamiento de información.

Los componentes del servicio OpenLDAP son:

- **slapd**, programa y herramientas encargadas del acceso a los datos.
- **Librerías** que implementan el protocolo LDAP.
- **Programas cliente**.

Para que dichos servicios funcionen correctamente están apoyados en una capa en segundo plano que es la encargada de almacenar y devolver los datos. Dicha capa de backend puede estar construida de distintas formas y es totalmente independiente de las herramientas desde las que se accede al servidor LDAP, de esta forma su utilización por diferentes servicios concurrentes es posible, suministrando así información a distintos consumidores de forma simultánea.

Openldap dispone de una serie de backends que se pueden dividir en tres categorías, aunque la inclusión de los mismos en una de ellas no lo exime de pertenecer a otra.

- **Almacenamiento de datos**, dichos backend se encargan de almacenar físicamente los datos. En este caso existen backends para bdb, hdb, ldif o nbd. Cada uno de ellos ofrece el acceso oportuno a las bases de datos que mantiene detrás.
- **Proxy backend**, en este caso el backend hace de puerta de enlace hacia otro sistema de almacenamiento. Aquí se encuentras las puertas hacia otros ldap, hacia archivos con metainformación, archivo */etc/passwd* de los sistemas UNIX, u otros sistemas como base de datos SQL.

- **Dinámicos**, encargados de generar datos dinámicamente, podemos encontrar backend para configuraciones, como buscadores de servicios, monitores de utilización, etc.

Además de los backends, OpenLDAP ofrece la posibilidad de insertar piezas de software entre el frontend y el backend, dichas piezas son denominadas overlays. Existen multitud de herramientas overlays que extienden la funcionalidad de OpenLDAP, haciéndolo un sistema modular, capaz de acoplarse a los entornos más exigentes. Dicha característica fue introducida en la versión 2.2 de OpenLDAP y en la actualidad cuenta con más de 20 herramientas implementadas en el núcleo del software. Dichas herramientas pueden realizar operaciones de registro de actividades, integración con otro tipo de servicios, transformación de las respuestas a otros protocolos, firma de operaciones, etc.

Ventajas

- Basado en LDAP lo que le hace partícipe de las ventajas de éste.
- Dispone de diversas herramientas que lo hacen accesible a través de la red, y con posibilidad de encriptación de los datos.
- La importación y exportación de elementos es sencilla y compatible con cualquier software que implemente el protocolo LDAP.
- Es de código abierto y libre de usar sin necesidad de adquirir licencias.

Desventajas

- Protocolo de manejo de datos poco intuitivo, pero existen múltiples herramientas que facilitan su uso.

2.3.3. Active Directory

☞ http://msdn.microsoft.com/en-us/library/windows/desktop/aa362244(v=vs.85).aspx

Active Directory (AD) es el término que usa Microsoft para referirse a su implementación de servicio de directorio en una red distribuida de computadores. Utiliza distintos protocolos LDAP, DNS, Dynamic Host Configuration Protocol (DHCP), etc. Su estructura jerárquica permite mantener una serie de objetos relacionados con componentes de una red, como usuarios, grupos de usuarios, permisos y asignación de recursos y políticas de acceso. AD permite a los administradores establecer políticas a nivel de empresa, desplegar programas en muchos ordenadores y aplicar actualizaciones críticas a una organización entera. AD almacena información de una organización en una base de datos central, organizada y accesible. Pueden encontrarse desde directorios con cientos de objetos para una red pequeña hasta directorios con millones de objetos.

Todos los componentes que forman parte del AD son denominados objetos, los cuales están estructurados jerárquicamente. Los objetos se enmarcan en tres grandes categorías: recursos, servicios y usuarios. El AD proporciona información sobre los objetos, los organiza, controla el acceso y establece la seguridad.

Cada objeto representa una entidad individual y sus atributos. Los objetos pueden contener otros objetos. Un objeto está unívocamente identificado por su nombre y tiene un conjunto de atributos - las características e información que el objeto puede contener - definidos por y dependientes del tipo. Los atributos, la estructura básica del objeto, se definen por un esquema, que también determina la clase de objetos que se pueden almacenar en el AD.

Ventajas

- Su compatibilidad con la mayoría de dispositivos está asegurada ya que es utilizado por la gran mayoría de las grandes empresas.

- Ofrece una interfaz de administración fácil para el usuario normal a través de las herramientas proporcionadas por Microsoft.

Desventajas

- Requiere su utilización con un servidor Microsoft Windows, lo cual encarece su uso.

2.4. Otros servicios

Además de los servicios anteriormente citados, un servidor para esta finalidad debe componerse de multitud de elementos software que en conjunto ofrecen la funcionalidad para la que se está realizando dicho proyecto. A continuación se enumerarán distintos servicios y software que en conjunto con los anteriores dotarán al servidor de una mayor complejidad.

2.4.1. Servidor web

Un servidor web o servidor Hypertext Transfer Protocol (HTTP) es un programa informático que procesa una aplicación del lado del servidor realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente. El código recibido por el cliente suele ser compilado y ejecutado por un navegador web. Para la transmisión de todos estos datos suele utilizarse algún protocolo. Generalmente se utiliza el protocolo HTTP para estas comunicaciones, perteneciente a la capa de aplicación del modelo OSI. El término también se emplea para referirse al ordenador que ejecuta el programa.

El Servidor web se ejecuta en un ordenador manteniéndose a la espera de peticiones por parte de un cliente y que responde a estas peticiones adecuadamente, mediante una página web que se exhibirá en el navegador, una respuesta a la invocación de un servicio o mostrando el respectivo mensaje si se detectó algún error. El servidor responde al cliente enviando el código, comúnmente en lenguaje HyperText Markup Language (HTML) de la página; el cliente, una vez recibido el código, lo interpreta y lo exhibe en pantalla. El cliente es el encargado de interpretar el código, es decir, de mostrar las fuentes, los colores y la disposición de los textos y objetos de la página; el servidor tan sólo se limita a transferir el código de la página sin llevar a cabo ninguna interpretación de la misma.

Además de la transferencia de código HTML, los servidores web pueden entregar aplicaciones web. Éstas son porciones de código que se ejecutan cuando se realizan ciertas peticiones o respuestas HTTP. Hay que distinguir entre:

- Aplicaciones en el lado del cliente: el cliente web es el encargado de ejecutarlas en la máquina del usuario. Suelen ser aplicaciones que ejecutan código Javascript: el servidor proporciona el

código de las aplicaciones al cliente y éste, mediante el navegador, las ejecuta. Es necesario, por tanto, que el cliente disponga de un navegador con capacidad para ejecutar aplicaciones (también llamadas scripts). Comúnmente, los navegadores permiten ejecutar aplicaciones escritas en lenguaje javascript, aunque pueden añadirse más lenguajes mediante el uso de plugins.

- Aplicaciones en el lado del servidor: el servidor web ejecuta la aplicación; ésta, una vez ejecutada, genera cierto código; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo HTTP.

Las aplicaciones de servidor muchas veces suelen ser la mejor opción para realizar aplicaciones web. La razón es que, al ejecutarse ésta en el servidor y no en la máquina del cliente, éste no necesita ninguna capacidad añadida, como sí ocurre en el caso de querer ejecutar aplicaciones javascript. Así pues, cualquier cliente dotado de un navegador web básico puede utilizar este tipo de aplicaciones.

Apache Web Server

☞ http://httpd.apache.org/

El servidor HTTP Apache es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Mac OS y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Además Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. En inglés, a patchy server (un servidor parcheado) suena igual que Apache Server.

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation. Apache presenta entre otras características el ser un servidor web altamente configurable, compatibilidad con conexión a distintas bases de datos de identificación y negociado de contenido, pero es criticado por la falta de una interfaz gráfica que ayude en su configuración.

Entre sus ventajas podemos enumerar las siguientes:

- Modular y por lo tanto extensible.
- De código abierto y multiplataforma.

- Muy configurable.

IIS

☞ http://www.iis.net/

Internet Information Services (IIS) es un servidor web y un conjunto de servicios para el sistema operativo Microsoft Windows. Originalmente era parte del Option Pack para Windows NT. Luego fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de IIS. Entre otros, los servicios que ofrece son: File Transfer Protocol (FTP), SMTP y HTTP/Secure Hypertext Transfer Protocol (HTTPS).

Antiguamente se denominaba Personal Web Server (PWS), y actualmente forma parte de la distribución estándar de Windows, de modo que no se necesita una licencia extra para instalarlo. Este servicio dota a un ordenador de las características de un servidor web para Internet o una intranet, es decir que pueden publicar páginas web tanto local como remotamente.

Los servicios de IIS proporcionan las herramientas y funciones necesarias para administrar de forma sencilla un servidor web seguro. El servidor web se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas. Por ejemplo, Microsoft incluye los de Active Server Pages (ASP) y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP o Perl.

Entre sus ventajas podemos destacar:

- Modular y por lo tanto extensible.
- Posee una interfaz de administración user friendly.

Nginx

☞ http://nginx.org/en/

Nginx es un servidor web y proxy inverso ligero de alto rendimiento y un proxy para protocolos de correo electrónico POP e IMAP.

Es software libre y de código abierto, licenciado bajo la Licencia BSD simplificada. Es multiplataforma, por lo que corre en sistemas tipo Unix (GNU/Linux, BSD, Solaris, Mac OS, etc.) y Microsoft Windows.

El servidor nginx es capaz de servir contenido HTTP tanto estático como dinámicamente, apoyándose en módulos de terceros capaces de gestionar las peticiones de interpretación de los distintos lenguaje utilizados en las páginas web, como pueden ser PHP o Perl. Está dirigido a servidores con una alta carga de peticiones, ya que la manera de asignar y responder las peticiones se realiza de forma asíncrona y dependiendo de los eventos generados. De esta forma los recursos no quedan bloqueados a nivel de proceso, como puede ocurrir con otro tipo de servidores web.

Las características y ventajas de este servidor se pueden resumir en:

- Concebido para soportar módulos que le den mayor funcionalidad.
- Soporte para FastCGI con opciones de caché.
- Habilitado para soportar un gran número de conexiones consumiendo pocos recursos.

2.4.2. Sistemas de gestión de bases de datos

El propósito general de los Sistema de Gestión de Base de Datos (SGBD) es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información.

Existen distintos objetivos que deben cumplir los SGBD:

- **Abstracción de la información**. Los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario. Así, se definen varios niveles de abstracción.
- **Independencia**. La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
- **Consistencia**. En aquellos casos en los que no se ha logrado eliminar la redundancia, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente,

es decir, que todos los datos repetidos se actualicen de forma simultánea. Por otra parte, la base de datos representa una realidad determinada que tiene determinadas condiciones, por ejemplo que los menores de edad no pueden tener licencia de conducir. El sistema no debería aceptar datos de un conductor menor de edad. En los SGBD existen herramientas que facilitan la programación de este tipo de condiciones.

- **Seguridad**. La información almacenada en una base de datos puede llegar a tener un gran valor. Los SGBD deben garantizar que esta información se encuentra segura y ofrece la posibilidad de asignar permisos a usuarios y grupos de usuarios, que permiten otorgar diversos niveles de acceso.

- **Manejo de transacciones**. Una transacción es un programa que se ejecuta como una sola operación. Esto quiere decir que luego de una ejecución en la que se produce una falla, el resultado es el mismo que se obtendría si el programa no se hubiera ejecutado. Los SGBD proveen mecanismos para programar las modificaciones de los datos de una forma mucho más simple que si no se dispusiera de ellos.

- **Tiempo de respuesta**. Lógicamente, es deseable minimizar el tiempo que el SGBD demora en proporcionar la información solicitada y en almacenar los cambios realizados.

Mysql

☞ http://www.mysql.com/

MySQL es un SGBD relacional, multihilo y multiusuario. Está desarrollado por los laboratorios MySQL AB, que pertenecen a la compañía Oracle, como software libre bajo un esquema de licenciamiento dual.

Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

MySQL es muy utilizado en aplicaciones web, como Drupal o phpBB, en plataformas (GNU/-Linux / Microsoft Windows - Apache - MySQL - PHP / Perl / Python), y por herramientas de

seguimiento de errores como Bugzilla. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL.

MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones. Sea cual sea el entorno en el que va a utilizar MySQL, es importante monitorizar de antemano el rendimiento para detectar y corregir errores tanto de SQL como de programación

Entre las principales características únicas de MySQL cabe destacar:

- Permite escoger entre múltiples motores de almacenamiento para las tablas. Dichos motores pueden ser desde archivos cvs, hasta motores desarrollados por terceros.
- Agrupación de transacciones entre distintas conexiones, para aumentar el número de transacciones por segundo.

Mongo DB

☞ http://www.mongodb.org/

MongoDB es un sistema de base de datos NoSQL orientado a documentos, desarrollado bajo el concepto de código abierto. MongoDB forma parte de la nueva familia de sistemas de base de datos NoSQL. En vez de guardar los datos en tablas como se hace en las base de datos relacionales, MongoDB guarda estructuras de datos en documentos tipo JSON con un esquema dinámico (MongoDB llama ese formato BSON), haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida. El código binario está disponible para los sistemas operativos Microsoft Windows, GNU/Linux, Mac OS y Unix.

Cabe destacar las siguientes características:

- Soporta la búsqueda por campos, consultas de rangos y expresiones regulares. Las consultas pueden devolver un campo específico del documento pero también puede ser una función JavaScript definida por el usuario.

- Cualquier campo en un documento de MongoDB puede ser indexado, al igual que es posible hacer índices secundarios. El concepto de índices en MongoDB es similar a los encontrados en base de datos relacionales.
- Soporta el tipo de replicación maestro-esclavo.
- Se puede escalar de forma horizontal usando el concepto de *shard*.
- Puede ser utilizado con un sistema de archivos, tomando la ventaja de la capacidad que tiene MongoDB para el balanceo de carga y la replicación de datos utilizando múltiples servidores para el almacenamiento de archivos.
- La función MapReduce puede ser utilizada para el procesamiento por lotes de datos y operaciones de agregación. Esta función permite que los usuarios puedan obtener el tipo de resultado que se obtiene cuando se utiliza el comando SQL *group-by*.
- Tiene la capacidad de realizar consultas utilizando JavaScript, haciendo que estas sean enviadas directamente a la base de datos para ser ejecutadas.

PostgreSQL

☞ http://www.postgresql.org/

PostgreSQL es un SGBD relacional orientado a objetos y libre, publicado bajo la licencia BSD. Dada su licencia BSD, se permite la utilización del código para ser comercializado. Uno de los casos ejemplo es la de Enterprise DB (Postgresql Plus), la cual incluye varias extensiones y una interfaz de desarrollo basada en Java.

Entre las principales características de este SGBD se puede destacar:

- Mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos.
- PostgreSQL provee de forma nativa soporte para una amplia variedad de tipos como puede ser: número de precisión arbitraria, texto de longitud ilimitada, figuras geométricas, direcciones MAC, etc. Incluso los usuarios pueden definir sus propios tipos de datos.

- Posibilidad de ejecutar funciones en diferentes lenguajes de programación como puede ser C, C++, Java, etc.

2.4.3. Servidor DNS

DNS es un sistema de nomenclatura jerárquica para ordenadores, servicios o cualquier recurso conectado a Internet o a una red privada. Este sistema asocia información variada con nombres de dominios asignado a cada uno de los participantes. Su función más importante, es resolver nombres inteligibles para las personas en identificadores binarios asociados con los equipos conectados a la red, esto con el propósito de poder localizar y direccionar estos equipos mundialmente.

El servidor DNS utiliza una base de datos distribuida y jerárquica que almacena información asociada a nombres de dominio en redes como Internet. Aunque como base de datos el DNS es capaz de asociar diferentes tipos de información a cada nombre, los usos más comunes son la asignación de nombres de dominio a direcciones IP y la localización de los servidores de correo electrónico de cada dominio.

La asignación de nombres a direcciones IP es ciertamente la función más conocida del protocolo DNS. Además de ser más fácil de recordar, el nombre es más fiable. La dirección numérica podría cambiar por muchas razones, sin que tenga que cambiar el nombre.

Para la operación práctica del sistema DNS se utilizan tres componentes principales:

- **Los Clientes fase 1**: Un programa cliente DNS que se ejecuta en la computadora del usuario y que genera peticiones DNS de resolución de nombres a un servidor DNS (Por ejemplo: ¿Qué dirección IP corresponde a nombre.dominio?);

- **Los Servidores DNS**: Que contestan las peticiones de los clientes. Los servidores recursivos tienen la capacidad de reenviar la petición a otro servidor si no disponen de la dirección solicitada.

- **Las Zonas de autoridad**, porciones del espacio de nombres raros de dominio que almacenan los datos. Cada zona de autoridad abarca al menos un dominio y posiblemente sus subdominios, si estos últimos no son delegados a otras zonas de autoridad

Djbdns

☞ http://cr.yp.to/djbdns.html

El paquete de software Djbdns es una implementación del servicio DNS realizada por Daniel J. Bernstein. Dicho paquete de software posee las herramientas necesarias para la creación y el mantenimiento de un servidor DNS.

Dichas herramientas se dividen en:

- **Herramientas de servidor**
  - dnscache, encargado de la resolución y de la caché.
  - tinydns, encargado de mantener una base de datos DNS.
  - walldns, que ofrece funcionalidad para llamadas inversas.
  - rbldns, funcionalidad para listas negras.
- **Herramientas de cliente**
  - axfr-get, herramienta para transferir configuraciones de zonas de DNS
  - otras herramientas DNS, como son gnsip, dnsipq, dnsname, etc. Utilizadas para gestionar y/o comprobar la configuración del servidor.

Bind

☞ https://www.isc.org/software/bind

BIND es el servidor de DNS más comúnmente usado en Internet, especialmente en sistemas Unix, en los cuales es un Estándar de facto. Es patrocinado por la Internet Systems Consortium.

Una nueva versión de BIND (BIND 9) fue escrita desde cero en parte para superar las dificultades arquitectónicas presentes anteriormente para auditar el código en las primeras versiones de BIND, y también para incorporar DNSSEC (DNS Security Extensions). Es comúnmente usado en sistemas GNU/Linux.

BIND 9 incluye entre otras características importantes:

- TSIG, protocolo de intercambio de firma.
- Notificación DNS.
- nsupdate.
- Soporte para IPv6.
- rndc flush, que refresca la cache de BIND.
- Vistas.
- Procesamiento en paralelo,
- Arquitectura enfocada hacia la portabilidad.

2.4.4. Autoridad de certificación

En criptografía una Certification Authority (CA) es una entidad de confianza, responsable de emitir y revocar los certificados digitales o certificados, utilizados en la firma electrónica, para lo cual se emplea la criptografía de clave pública.

Modo de funcionamiento

1. **Solicitud de un certificado.** El mecanismo habitual de solicitud de un certificado a una CA consiste en que la entidad solicitante, utilizando ciertas funciones del software de criptografía, completa ciertos datos identificativos (entre los que se incluye el nombre del servidor) y genera una pareja de claves pública/privada. Con esa información el software de servidor compone un fichero que contiene una petición Certificate Signing Request (CSR) en formato PKCS#10 que contiene la clave pública y que se hace llegar a la CA elegida. Esta, tras verificar por sí o mediante los servicios de una RA (Registration Authority, Autoridad de Registro) la información de identificación aportada, envía el certificado firmado al solicitante, que lo instala en el servidor web con la misma herramienta con la que generó la petición CSR.

2. **Jerarquía de certificación.** Las CA disponen de sus propios certificados públicos, cuyas claves privadas asociadas son empleadas por las CA para firmar los certificados que emiten. Un

certificado de CA puede estar auto-firmado cuando no hay ninguna CA de rango superior que lo firme. Este es el caso de los certificados de CA raíz, el elemento inicial de cualquier jerarquía de certificación. Una jerarquía de certificación consiste en una estructura jerárquica de CAs en la que se parte de una CA auto-firmada, y en cada nivel, existe una o más CAs que pueden firmar certificados de entidad final (titular de certificado: servidor web, persona, aplicación de software) o bien certificados de otras CA subordinadas plenamente identificadas y cuya Política de Certificación sea compatible con las CAs de rango superior.

3. **Confianza en la CA.** Una de las formas por las que se establece la confianza en una CA para un usuario consiste en la instalación en el ordenador del usuario (tercero que confía) del certificado autofirmado de la CA raíz de la jerarquía en la que se desea confiar. Si está instalada una CA en el repositorio de CAs de confianza de cada navegador, cualquier certificado firmado por dicha CA se podrá validar, ya que se dispone de la clave pública con la que verificar la firma que lleva el certificado. Cuando el modelo de CA incluye una jerarquía, es preciso establecer explícitamente la confianza en los certificados de todas las cadenas de certificación en las que se confíe. Para ello, se puede localizar sus certificados mediante distintos medios de publicación en Internet, pero también es posible que un certificado contenga toda la cadena de certificación necesaria para ser instalado con confianza.

Openssl

☞ http://www.openssl.org/

OpenSSL es un proyecto de software libre basado en SSLeay. Consiste en un robusto paquete de herramientas de administración y bibliotecas relacionadas con la criptografía, que suministran funciones criptográficas a otros paquetes como *OpenSSH* y navegadores web.

Estas herramientas ayudan al sistema a implementar el Secure Socket Layer (SSL), así como otros protocolos relacionados con la seguridad, como el Transport Layer Security (TLS). Este paquete de software es importante para cualquiera que esté planeando usar cierto nivel de seguridad en su máquina con un sistema operativo libre basado en GNU/Linux. OpenSSL también permite crear certificados digitales que pueden aplicarse a un servidor.

2.4.5. Lenguajes de programación web

Los lenguajes de programación del lado del servidor es una tecnología que consiste en el procesamiento de una petición de un usuario mediante la interpretación de un script en el servidor web para generar páginas HTML dinámicamente como respuesta.

PHP

☞ http://php.net/

PHP es un lenguaje de programación de uso general de que se ejecuta del lado del servidor, originalmente fue diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante. PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. PHP puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas. Se distribuye bajo su propia licencia PHP, que es muy similar a la licencia GNU.

Entre las características de PHP se pueden destacar:

- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- El código fuente escrito en PHP es invisible al navegador web y al cliente, ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad.
- Capacidad de expandir su potencial utilizando módulos (llamados ext's o extensiones).

- Posee una amplia documentación en su sitio web oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso universal.
- Permite aplicar técnicas de programación orientada a objetos.
- Tiene manejo de excepciones (desde la versión 5).

Python

☞ http://www.python.org/

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis limpia y que favorece la lectura del código fuente. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma. Posee una licencia de código abierto, denominada Python Software Foundation License que es compatible con la Licencia pública general de GNU a partir de la versión 2.1.1, e incompatible en ciertas versiones anteriores.

Python es un lenguaje de programación multiparadigma. Esto significa que más que forzar a los programadores a adoptar un estilo particular de programación, permite varios estilos: programación orientada a objetos, programación imperativa y programación funcional. Otros paradigmas están soportados mediante el uso de extensiones. Python usa tipado dinámico y conteo de referencias para la administración de memoria.

Una característica importante de Python es la resolución dinámica de nombres; es decir, lo que enlaza un método y un nombre de variable durante la ejecución del programa (también llamado enlace dinámico de métodos).

Otro objetivo del diseño del lenguaje es la facilidad de extensión. Se pueden escribir nuevos módulos fácilmente en C o C++. Python puede incluirse en aplicaciones que necesitan una interfaz programable.

Python es usado como lenguaje de script para aplicaciones web. Posee un gran número de frameworks de desarrollo que facilitan a los desarrolladores el diseño y mantenimiento de aplicaciones complejas. Además hay desarrolladas multitud de librerías con distintos propósitos, desde modelado en 3D, librerías para cálculos matemáticos, o utilizadas en inteligencia artificial.

ASP.NET

☞ http://www.asp.net/

ASP.NET es un framework para aplicaciones web desarrollado y comercializado por Microsoft. Es usado por programadores para construir sitios web dinámicos, aplicaciones web y servicios web XML. Apareció en enero de 2002 con la versión 1.0 del .NET Framework, y es la tecnología sucesora de la tecnología Active Server Pages (ASP). ASP.NET esta construido sobre el Common Language Runtime, permitiendo a los programadores escribir código ASP.NET usando cualquier lenguaje admitido por el .NET Framework

Características:

- Las páginas de ASP.NET, conocidas oficialmente como formularios web, son el principal medio de construcción para el desarrollo de aplicaciones web.
- ASP.NET sólo funciona sobre el servidor de Microsoft IIS, lo que supone una desventaja respecto a otros lenguajes que se ejecutan en el servidor.
- Microsoft recomienda que para realizar programación dinámica se use el modelo code-behind, o de respaldo, que coloca el código en un archivo separado o en una etiqueta de script especialmente diseñada.
- ASP.NET permite la creación de componentes reutilizables a través de la creación de Controles de Usuario (User Controls).
- Posee una amplia gama de extensiones que aumentan la funcionalidad de las páginas web creadas con dicho lenguaje. Se pueden destacar entre otras ASP.NET AJAX, ASP.NET MVC Framework, etc.

2.4.6. Cortafuegos, seguridad perimetral

Un cortafuegos es una parte de un sistema o una red que está diseñada para bloquear el acceso no autorizado, permitiendo al mismo tiempo comunicaciones autorizadas.

Se trata de un dispositivo o conjunto de dispositivos configurados para permitir, limitar, cifrar, descifrar, el tráfico entre los diferentes ámbitos sobre la base de un conjunto de normas y otros criterios.

Los cortafuegos pueden ser implementados en hardware o software, o una combinación de ambos. Los cortafuegos se utilizan con frecuencia para evitar que los usuarios de Internet no autorizados tengan acceso a redes privadas conectadas a Internet, especialmente intranets. Todos los mensajes que entren o salgan de la intranet pasan a través del cortafuegos, que examina cada mensaje y bloquea aquellos que no cumplen los criterios de seguridad especificados. También es frecuente conectar al cortafuegos a una tercera red Demilitarized Zone (DMZ), en la que se ubican los servidores de la organización que deben permanecer accesibles desde la red exterior.

La seguridad perimetral corresponde a la integración de elementos y sistemas electrónicos, para la protección de perímetros, detección de tentativas de intrusión y/o disuasión de intrusos en servicios especialmente sensibles.

Iptables

☞ http://www.netfilter.org/projects/iptables/index.html

IPTables es el programa mediante la cual el administrador puede definir políticas de filtrado del tráfico que circula por la red. El nombre IPTables se utiliza frecuentemente de forma errónea para referirse a toda la infraestructura ofrecida por el proyecto Netfilter. Sin embargo, este proyecto ofrece otros subsistemas independientes de IPTables tales como el connection tracking system o sistema de seguimiento de conexiones, que permite encolar paquetes para que sean tratados desde espacio de usuario. IPTables es un software disponible en prácticamente todas las distribuciones de GNU/Linux actuales.

Hay que tener en cuenta que IPTables forma parte del núcleo de GNU/Linux. Eso quiere decir que el comando IPTables sólo modifica la configuración del cortafuegos de GNU/Linux. Eso significa que el cortafuegos no puede ser ni apagado ni encendido, ni tampoco desinstalado o actualizado. Para actualizarlo hay que actualizar el núcleo. Este aislamiento es ideal para un cortafuegos, ya que así ninguna aplicación maliciosa podrá alterar su comportamiento en su beneficio.

IPTables está compuesto por:

- **Reglas**: Permite filtrar un paquete y decidir si dejarlo pasar o no. Forma parte siempre de un listado que se llama Cadena.

- **Cadenas**: Es el listado de reglas. Éstas se ejecutan de arriba a abajo hasta que se cumpla una de ellas. La Cadena tiene lo que se le llama Política, que sirve para decidir qué hacer con los paquetes que no coincidieron con ninguna de las reglas. A cada cadena le entran ciertos paquetes, dependiendo de la naturaleza de la cadena.

- **Tablas**: Es la encargada de contener las cadenas. Generalmente suelen haber 3 cadenas por tabla. Cada tabla tiene funciones específicas. Existen unas tablas de sistema que vienen de serie y no se pueden quitar. Éstas son: *filter*, *nat* y *mangle*.

Cada cadena representa un proceso distinto de un paquete en el cortafuegos, por lo que un paquete puede llegar a pasar por dos o más cadenas en momentos distintos.

Todas las cadenas tienen lo que se llama política: especifica qué hacer con los paquetes con los que las reglas de la propia cadena no supo qué hacer [4].

Se pueden resumir las distintas tablas de IPTables de la siguiente forma:

- **Tabla de filtros**: es la más básica y la que se usa por defecto. Su funcionalidad consiste en interceptar todos los paquetes cuando entran o salen de la máquina, tenemos tres cadenas dentro:

  - **INPUT**: Intercepta todos los paquetes que entran a los dispositivos de red.
  - **OUTPUT**: Intercepta todos los paquetes de red que genera la máquina.
  - **FORWARD**: Intercepta todos los paquetes que intentan cruzar entre una red y otra.

- **Tabla de traducción de direcciones de red**: Esta tabla es la responsable de configurar las reglas de reescritura de direcciones o de puertos de los paquetes.

  - **PREROUTING**: Los paquetes entrantes pasan a través de esta cadena antes de que se consulte la tabla de rutas locales.
  - **POSTROUTING**: Los paquetes salientes pasan por esta cadena después de haberse tomado la decisión de la ruta a seguir.

- **OUTPUT**: Cadena de salida.

- **Tabla de destrozo**: Esta tabla es la responsable de ajustar las opciones de los paquetes, como por ejemplo la calidad de servicio. Todos los paquetes pasan por esta tabla. Debido a que está diseñada para efectos avanzados, contiene todas las cadenas predefinidas posibles:

  - **PREROUTING**: Todos los paquetes que logran entrar a este sistema, antes de que la tabla de rutas decida si el paquete debe ser reenviado o si tiene destino local.
  - **INPUT**: Todos los paquetes destinados para este sistema pasan a través de esta cadena.
  - **FORWARD**: Todos los paquetes que exactamente pasan por este sistema pasan a través de esta cadena.
  - **POSTROUTING**: Todos los paquetes creados en este sistema pasan a través de esta cadena.
  - **OUTPUT**: Todos los paquetes que abandonan este sistema pasan a través de esta cadena.

En la figura 2.1 se puede observar el recorrido que hace un paquete al entrar en las tablas de IPTables.

Todas y cada una de las reglas tiene definido un destino definido por el usuario. Por defecto existen varios destinos incorporados en el kernel, aunque se pueden definir un mayor número de ellos gracias a las extensiones. Cuando un paquete es recogido por una regla, es enviado al destino especificado, donde el programa o extensión correspondiente lo procesa. Los siguientes destinos son los definidos por defecto por IPTables:

- **ACCEPT**: El paquete es aceptado y será procesado según la tabla y/o cadena de la que provenga.

- **DROP**: El paquete es rechazado, y no se realiza ningún otro procesamiento sobre él.

- **QUEUE**: El paquete se encola, y se deja para procesar por otras reglas o bibliotecas externas. Si no se realiza ninguna acción, el paquete termina siendo descartado.

- **RETURN**: El paquete no circulará por la cadena que lo capturó. En caso de cadenas anidadas, el paquete volverá a la cadena padre.

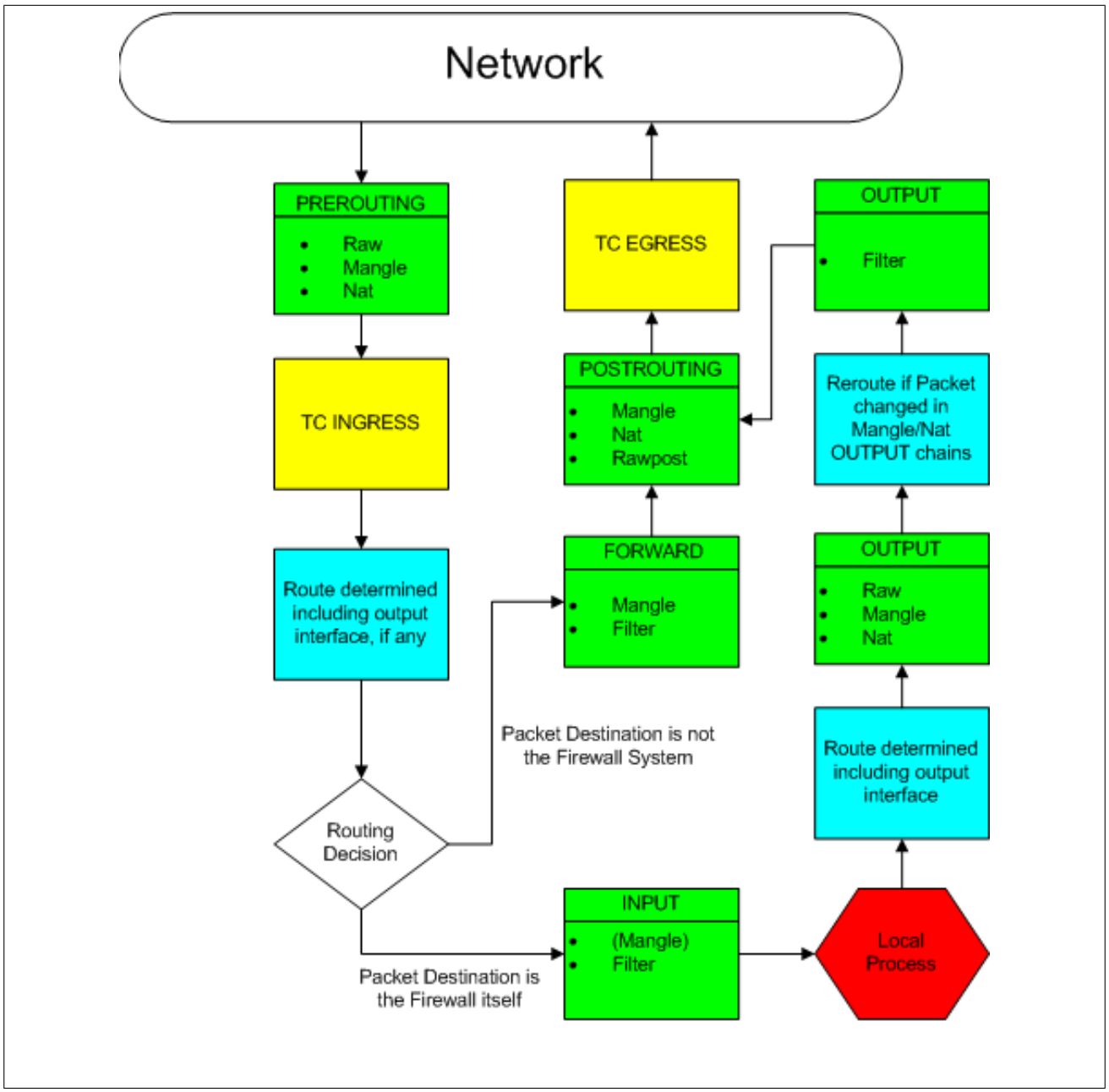

Figura 2.1: *Flujo de paquetes para Iptables [2].*

Estos, son algunos de los destinos que se pueden encontrar en las extensiones más populares.

- **REJECT**: Al igual que *DROP*, en este caso el paquete es rechazado, pero se envía un paquete de error a quien lo envió originalmente.
- **LOG**: Las reglas que han cumplido los paquetes son almacenados en un fichero de log para su análisis.
- **ULOG**: Al igual que el anterior, los paquetes serán utilizados para llevar un log de lo ocurrido, pero a diferencia, dicho destino tendrá abierto un socket para que distintos programas puedan puedan conectarse a él.
- **DNAT**: En este caso la dirección y/o el puerto de destino podrán ser reescritos para traducir su dirección de red.
- **SNAT**: En este caso la dirección y/o el puerto de origen podrán ser reescritos para traducir su dirección de red.

2.4.7. Antivirus y Antispam

Las aplicaciones que se detallan a continuación sirven para proteger tanto al usuario como a los servicios contra el ataque de programas maliciosos. En este caso existen dos tipos de aplicación que harán dicha labor, el antivirus, encargado de escanear los correos enviados y recibidos en busca de amenazas, y el antispam, encargado de decidir si los correos recibidos son legítimos y no contienen enlaces o publicidad no deseada.

En este caso el antivirus es la aplicación o grupo de aplicaciones dedicadas a la prevención, búsqueda, detección y eliminación de programas malignos en los correos. Entre los programas con códigos malignos se incluyen virus, troyanos, gusanos, spywares, entre otros malwares. Para que el antivirus sea considerado efectivo y eficiente deberá cumplir algunos de estos requisitos:

- Actualizaciones constantes.
- Protección permanente.
- Completa base de datos de programas malignos.

- Buena heurística.

En cuanto al antispam, es una aplicación o herramienta informática que se encarga de detectar y eliminar el spam y los correos no deseados. El principal objetivo de una herramienta antispam, es lograr un buen porcentaje de filtrado de correo no deseado. Pero tampoco deben identificar al correo deseado como no deseado, pues eso traería peores consecuencias que olvidar filtrar algún spam.

Las herramientas antispam utilizan múltiples técnicas para detectar el correo no deseado. Estas técnicas pueden resumirse en:

- **Técnicas manuales**
  - Emplear un diccionario propio para detectar palabras que suelen aparecer en estos correos. Ese diccionario puede ser armado con palabras que el propio administrador del sistema identifica como spam manualmente, o armado de forma inteligente por la aplicación, cuando el usuario selecciona qué es deseado y qué es no deseado de su bandeja de entrada.
  - Uso de una lista blanca y una lista negra. El programa o el administrador manualmente identifica las direcciones y nombres que son considerados para estar en la lista blanca, cuyos correos serán enviados, y cuales estarán en la lista negra, y por lo tanto no se recibirán sus correos.
- **Técnicas automáticas**
  - El servidor se encargan de analizar cada uno de los correos electrónicos que llegan al usuario, para identificar si son o no spam. Esos servidores remotos utilizan grandes bases de datos con información (direcciones IP, nombres, textos, etc.) para identificar el correo no deseado.

Se detallan como ejemplos de antivirus y software antispam dedicados a servidores de correos los siguientes programas informáticos.

ClamAV

☞ http://www.clamav.net/

ClamAV es un antivirus bajo licencia GPL diseñado para detectar troyanos, virus, software malicioso y otras amenazas sobre los sistemas informáticos. ClamAV nació como un proyecto de código abierto que pretende identificar y bloquear virus en el sistema. El primer objetivo de ClamAV fue combatir el correo electrónico con contenidos maliciosos. Como consecuencia de ello, ClamAV se está usando en un número elevado de servidores de correo electrónico.

Gracias a la colaboración de varias compañías, universidades y otras organizaciones le ha sido posible al proyecto ClamAV poseer una red extensa de distribución rápida y fiable en todo el mundo.

El objetivo primario de ClamAV es la consecución de un conjunto de herramientas que identifiquen y bloqueen el software malicioso proveniente del correo electrónico. Uno de los puntos fundamentales en este tipo de software es la rápida localización e inclusión en la herramienta de los nuevos virus encontrados y escaneados. Esto se consigue gracias a la colaboración de los miles de usuarios que usan ClamAV y a distintos sitios que proporcionan los virus escaneados.

Otra pieza clave de ClamAV es el soporte de desarrolladores que posee en todo el mundo; esta red de desarrolladores global posibilita una rápida reacción ante cualquier evidencia de un nuevo virus.

El proyecto ClamAV se desarrolla gracias a una red de contribuidores (proporcionan parches, información de errores, soporte técnico y documentación). Por otro lado, existe una serie de personas e instituciones que colaboran con donaciones a la realización del proyecto. Existe un comité de dirección que supervisa y coordina el proyecto siguiendo los patrones de La Catedral y el Bazar[18].

También cabe destacar la posibilidad de ejecución en distintos SO, así como poseer multitud de interfaces gráficas que lo hacen fácil de utilizar. Su efectividad está más que demostrada, ya que es utilizado en numerosos entornos empresariales y por una gran parte de los servidores de correo electrónico.

Amavis

☞ http://www.amavis.org/

Amavis es un filtro de contenido para correos electrónicos, liberado bajo licencia GPL. Su utilización se basa en la decodificación de los mensajes de correo, así como el procesamiento y análisis de los archivos adjuntos, basados en filtros externos que proporciona protección frente a correo no deseado, virus, y otro malware. Podría ser considerado como una interfaz entre el servicio MTA y los filtros correspondientes.

Amavis puede ser utilizado para:

- Detección de virus, correo electrónico no deseado y bloquear contenido potencialmente peligroso incluido en los mensajes de correo electrónico.
- Bloquear, marcar o redirigir correos dependiendo de su contenido.
- Bloquear mensajes en cuarentena para su posterior análisis.
- Generar firmas para identificación del dominio, DomainKeys Identified Mail (DKIM).
- Verificar dichas firmas contra listas blancas DKIM.

SpamAssassin

☞ http://spamassassin.apache.org/

SpamAssassin es un programa informático bajo licencia Apache License 2.0 utilizado para filtrar mensajes de correo, dicho filtrado se produce por la inclusión de reglas.

Dentro de estas reglas pueden destacarse:

- Detección de dirección basadas en DNS.
- Detección de la suma hash de los mensajes.
- Listas blancas y negras.
- Conexión con bases de datos en línea.

- Conexión con programa externos.
- Técnicas basadas en el teorema de Thomas Bayes.

El programa puede ser integrado fácilmente con un servidor de correo para filtrar todo el correo automáticamente. Además puede ser ejecutado por usuarios individuales para filtrar su propio buzón de correo e inclusive, conectarse con multitud de programas lectores de correo. Su facilidad de configuración y gran capacidad de extensión lo hacen uno de los programas software más utilizados para el filtrado de correo electrónico no deseado.

SpamAssassin está escrito en Perl y su funcionamiento es bastante sencillo. SpamAssassin contiene un gran número de reglas las cuales son aplicadas a cada uno de los correos electrónicos que debe filtrar. La mayoría de dichas reglas están basadas en expresiones regulares, que buscan coincidencias en el emisor, o en el cuerpo del mensaje.

Estas reglas son llamadas *test* por la documentación de SpamAssassin. Cada test devuelve una puntuación al mensaje, que al final de todos y cada uno de ellos da una nota al mensaje en cuestión. Si la nota supera unos límites predefinidos el mensaje es considerado no deseado. Con el correo no deseado pueden realizarse múltiples acciones, desde enviarlo al usuario marcado como no deseado, o enviarlo a un buzón especial, o simplemente ignorarlo. La configuración dependerá de cada uno de los usuarios.

Como su funcionamiento está basado en la heurística, puede ser que se filtren correos electrónicos que se hayan marcado de forma errónea como no deseados. Por eso también es recomendable el empleo de otras técnicas para marcar un correo como deseado o no deseado. Estas técnicas se han comentado al principio de la sección, y consiste en el empleo de listas de correos blancas y negras, detección por DNS, listas negras de Uniform Resource Identifiers (URIs) que pueden estar contenidas en los mensajes, etc.

2.4.8. Máquinas virtuales

En informática una máquina virtual es un software que simula a un ordenador y puede ejecutar programas como si fuese uno real. Este software en un principio fue definido como un duplicado eficiente y aislado de una máquina física. La acepción del término actualmente incluye a máquinas

virtuales que no tienen ninguna equivalencia directa con ningún hardware real.

Una característica esencial de las máquinas virtuales es que los procesos que ejecutan están limitados por los recursos y abstracciones proporcionados por ellas. Estos procesos no pueden escaparse de este espacio de hardware virtual.

Uno de los usos domésticos más extendidos de las máquinas virtuales es ejecutar sistemas operativos para probarlos. De esta forma podemos ejecutar un SO que queramos probar (GNU/Linux, por ejemplo) desde el SO habitual (Mac OS por ejemplo).

Existen dos tipos de máquinas virtuales:

- **Máquinas virtuales de proceso**, son aquellas que se ejecutan como un proceso dentro de un SO. Proporciona un entorno independiente de la plataforma hardware y del SO.

- **Máquinas virtuales de sistema**, son aquellas en las que se permite que la máquina física subyacente pueda multiplicarse y funcionar con varias máquinas virtuales, cada una de ellas ejecutando su propio SO. A la capa de software que permite esta virtualización se le denomina monitor de máquina virtual o hipervisor. Dicho hipervisor puede funcionar como un SO independiente, o residir como un proceso más de otro SO.

Para la realización de este proyecto se utilizarán este último tipo de máquinas virtuales, de las que se pueden destacar las siguientes características:

- Varios SO distintos pueden coexistir sobre el mismo hardware, en sólido aislamiento el uno del otro, por ejemplo para probar un SO nuevo sin necesidad de instalarlo directamente sobre el hardware.

- La máquina virtual puede proporcionar una arquitectura de instrucciones (ISA) que sea algo distinta de la verdadera máquina. Es decir, podemos simular hardware.

- Varias máquinas virtuales (cada una con su propio SO llamado sistema operativo invitado), pueden ser utilizadas para consolidar servidores. Esto permite que servicios que normalmente se tengan que ejecutar en computadoras distintas para evitar interferencias, se puedan ejecutar en la misma máquina de manera completamente aislada y compartiendo los recursos de una única computadora. La consolidación de servidores a menudo contribuye a reducir el coste

total de las instalaciones necesarias para mantener los servicios, dado que permiten ahorrar en hardware.

- La virtualización es una excelente opción hoy día, ya que las máquinas actuales (ordenadores portátiles, ordenadores de escritorio, servidores) en la mayoría de los casos están siendo subutilizados (gran capacidad de disco duro, memoria RAM, etc.), llegando a un uso de entre 30 % a 60 % de su capacidad. Al virtualizar, la necesidad de nuevas máquinas en una ya existente permite un ahorro considerable de los costes asociados (energía, mantenimiento, espacio, etc).

A continuación se detallarán algunos de los sistemas de máquinas virtuales más utilizados y extendidos.

Oracle VirtualBox

☞ https://www.virtualbox.org/

Oracle VM VirtualBox es un software de virtualización para arquitecturas x86/amd64, creado originalmente por la empresa alemana innotek GmbH. Actualmente es desarrollado por Oracle Corporation como parte de su familia de productos de virtualización. Por medio de esta aplicación es posible instalar sistemas operativos adicionales, conocidos como sistemas invitados, dentro de otro sistema operativo anfitrión, cada uno con su propio ambiente virtual.

Entre los sistemas operativos soportados (en modo anfitrión) se encuentran GNU/Linux, Mac OS X, OS/2 Warp, Microsoft Windows, y Solaris/OpenSolaris, y dentro de ellos es posible virtualizar los sistemas operativos FreeBSD, GNU/Linux, OpenBSD, OS/2 Warp, Microsoft Windows, Solaris, MS-DOS y muchos otros.

La aplicación fue inicialmente ofrecida bajo una licencia de software privativo, pero en enero de 2007, después de años de desarrollo, surgió VirtualBox OSE (Open Source Edition) bajo la licencia GPL 2. Actualmente existe la versión privativa Oracle VM VirtualBox, que es gratuita únicamente bajo uso personal o de evaluación, y esta sujeta a la licencia de Uso Personal y de Evaluación VirtualBox y la versión Open Source, VirtualBox OSE, que es software libre, sujeta a la licencia GPL.

VirtualBox ofrece algunas funcionalidades interesantes, como la ejecución de maquinas virtuales de forma remota, por medio del Remote Desktop Protocol (RDP), soporte iSCSI, aunque estas opciones no están disponibles en la versión OSE.

En cuanto a la emulación de hardware, los discos duros de los sistemas invitados son almacenados en los sistemas anfitriones como archivos individuales en un contenedor llamado Virtual Disk Image, incompatible con los demás software de virtualización.

Otra de las funciones que presenta es la de montar imágenes ISO como unidades virtuales ópticas de CD o DVD, o como un disquete.

Tiene un paquete de controladores que permiten aceleración en 3D, pantalla completa, hasta 4 placas PCI Ethernet (8 si se utiliza la línea de comandos para configurarlas), integración con teclado y ratón, etc.

VMWare

☞ http://www.vmware.com/

VMWare es un sistema de virtualización por software. Un sistema virtual por software es un programa que simula un sistema físico (un ordenador, un hardware) con unas características determinadas. Cuando se ejecuta el programa (hipervisor), proporciona un ambiente de ejecución similar a todos los efectos a un ordenador físico (excepto en el puro acceso físico al hardware simulado), con CPU (puede ser más de una), BIOS, tarjeta gráfica, memoria RAM, tarjeta de red, sistema de sonido, conexión USB, disco duro, etc.

Un virtualizador por software permite ejecutar (simular) varios ordenadores (SO) dentro de un mismo hardware de manera simultánea, permitiendo así el mayor aprovechamiento de recursos. No obstante, y al ser una capa intermedia entre el sistema físico y el sistema operativo que funciona en el hardware emulado, la velocidad de ejecución de este último es menor, pero en la mayoría de los casos suficiente para usarse en entornos de producción.

El rendimiento del sistema virtual varía dependiendo de las características del sistema físico en el que se ejecute, y de los recursos virtuales (CPU, RAM, etc.) asignados al sistema virtual.

VMWare virtualiza la plataforma sobre la que está ejecutando, de forma que la mayor parte de las instrucciones en VMWare se ejecutan directamente sobre el hardware físico, mientras que en

otros sistemas se traducen en llamadas al SO que se ejecuta en el sistema físico.

Entre las distintas versiones de software de virtualización de VMWare se puede destacar el servidor VMWare ESX y su homólogo libre y con limitaciones VMWare ESXi.

VMWare ESX es un sistema complejo de virtualización, pues corre como SO dedicado al manejo y administración de máquinas virtuales dado que no necesita un SO host sobre el cual sea necesario instalarlo. Pensado para la centralización y virtualización de servidores, esta versión no es compatible con una gran lista de hardware doméstico, por ejemplo no reconoce los disco IDE como unidades de almacenamiento y seria inútil instalarlo en este tipo de discos (en la versión 3.5 ya esta soportado sata). Es realmente útil, ya que solamente ocupa 10 Mb de Ram y 55 Mb de Disco Duro, aproximadamente... Para su administración, hay que instalar un software en una máquina remota, que se conecta por entorno web o consola de administración.

Hyper-v

☞ http://www.microsoft.com/hyper-v-server/

Microsoft Hyper-V es un programa de virtualización basado en un hipervisor para los sistemas de 64-bits con los procesadores basados en AMD-V o Tecnología de virtualización Intel (el instrumental de gestión también se puede instalar en sistemas x86). Una versión beta de Hyper-V se incluyó en el Microsoft Windows Server 2008 y la versión definitiva se publicó el 26 de junio de 2008.

La versión actual de Hyper-V, incluida en Microsoft Windows Server 2008 R2 como rol de servidor, agregó mejoras y nuevas funcionalidades como Live Migration, almacenamiento en máquinas virtuales dinámicas, y compatibilidad mejorada con procesadores y redes.

2.5. En este capítulo

En este capítulo se ha dado una visión general de todo lo que puede rodear a la consecución final del proyecto. Se ha hablado de los distintos SO, los servidores de correo actuales, servidores de directorio y otros servicios importantes que se deben tener en cuenta en un sistema de estas características.

Se han obviado los sistemas en línea que ofrecen funcionalidades similares como los populares Gmail, Yahoo, Outlook.com, etc, ya que tanto los datos como el funcionamiento del resto del sistema quedaba fuera del alcance del administrador.

En el siguiente capítulo se va a hablar sobre el sistema actual que está instalado en la empresa, así como todo el software que se encuentra. Además se hará un resumen de los sistemas escogidos para ser instalados y el porqué de la decisión.

3

Análisis

Este análisis contemplará el estado actual de los servicios y equipos que serán sustituidos o complementados a lo largo de la duración del proyecto.

3.1. Estado actual

La empresa actual cuenta en torno a 20 empleados, entre técnicos, comerciales, personal de gestión y administración. Para las comunicaciones de los distintos servicios utilizados por la empresa, se dispone de herramientas independientes y no sincronizables, lo que se traduce en una menor efectividad a la hora de consultar agendas, programar reuniones o intercambiar contactos. Además las distintas herramientas utilizadas, como pueden ser, herramientas de gestión de proyectos, herramientas de seguimiento de clientes, monitorización de servidores, etc. necesitan disponer de servidores de comunicación rápidos y fiables.

En este momento se cuenta con un servidor de correo interno, encargado de recibir y enviar los correos de dichos servicios. Dicho correo no está conectado al exterior debido a la falta de mantenimiento del mismo, así como el temor de la exposición del servidor a internet. Dicho miedo viene derivado de la posible infección del mismo por algún tipo de virus, o la introducción de software malicioso que comprometiese la seguridad empresarial. Además, la utilización del servidor para otros fines alejados del rol de servidor de correo, hace del mismo un punto de gran valor, que no se puede permitir que sea vulnerado.

El modo de conexión actual al servidor de correo, no permite el acceso desde redes externas a la oficina. De este modo cualquier consulta al correo que se tuviese que realizar desde fuera de la oficina, entraña una problemática que sólo podía resolverse mediante una conexión VPN. Además, la necesidad de actualización de la infraestructura y el aumento del número de servicios necesarios, hacían del sistema actual, un sistema obsoleto y con necesidades de renovación.

La máquina actual, un servidor Debian GNU/Linux versión Etch, no está siendo mantenida por problemas derivados de compatibilidad de las nuevas versiones con configuraciones anteriores que se hicieron, lo que hace inviable su actualización. Además dicha máquina se utiliza también como servidor proxy para Hypertext Transfer Protocol (HTTP) y servidor de ficheros File Transfer Protocol (FTP). Funciones que deberían estar totalmente separadas y en servidores independientes.

3.2. Estado de los sistemas

A continuación se estudiará en profundidad el diseño de los sistemas. Dichos sistemas cumplían parte de los requisitos del actual proyecto.

3.2.1. Hardware

El servidor actual se compone de las siguientes características hardware:

- Un procesador Intel Xeon 3040.
- 1Gb de memoria RAM distribuida en dos módulos.
- Dos discos de 250Gb en RAID 1.
- Doble interfaz de Red Gigabit.

Todo este hardware está montado sobre un servidor Dell Poweredge 860 y montado en un armario en la propia oficina. El mantenimiento físico de dicho servidor es un punto en contra del mismo, ya que los costes de energía y refrigeración son elevados para los servicios que presta.

3.2.2. Servicios

Los servicios ofrecidos por el antiguo sistema eran los siguientes:

- Sistema operativo (SO), Debian versión Etch, fue instalado en los comienzos de la empresa, en el año 2007. Está instalado sobre un servidor físico, por lo que la única manera de aprovechar totalmente su hardware es añadir más servicios a la misma máquina. De esta forma se incluyeron varios servicios que podrían comprometer el resto. Es por eso que no todos los servicios tiene comunicación con el exterior, haciendo imposible el acceso al servicio de correo desde internet. Además, la utilización de los servicios de dicho servidor tan sólo se puede realizar mediante usuarios y contraseñas exclusivos e independientes del resto, lo que hace un poco más difícil su administración.

- **Servidor de correo**, el servidor de correo está montado sobre los servicios Postfix en su versión 2.3 y Courier IMAP en su versión 4.1.

- **Servidor de base de datos**, Mysql 5.0, cuyo único fin es proveer la base de datos que utiliza postfix para su configuración. Como se ve es un recurso que está bastante desaprovechado.

- **Otro software**, además del anteriormente citado, comentar que el servidor posee de una serie de software a destacar:

  - Servidor FTP Pure-ftpd 1.0
  - Servidor HTTP ssub:apache 2.2
  - Antivirus ClamAV 0.90
  - Antivirus Amavis 2.4
  - Servidor Domain Name System (DNS) Djbdns

Se puede observar la falta de actualización de los sistemas, con versiones bastante antiguas y que hace que la búsqueda de nuevas características o ampliación de las funcionalidades sea una tarea imposible. Además la infrautilización del hardware y los servicios, hacen del sistema algo caro de mantener, ya que los recursos podrían utilizarse para ofrecer más funcionalidades.

Al ser un sistema sin virtualizar, la instalación de distintas herramientas puede comprometer al resto, por ejemplo, se puede observar que el sistema posee un servidor de FTP, el cuál puede comprometer al resto de servicios, ya que dicho protocolo no es uno de los más seguros de internet. Por lo tanto el servidor debe permanecer aislado totalmente de internet dada su fragilidad.

Por otra parte la instalación de servicios más actuales requerirán de ciertos programas en sus versiones más actuales. Dado que la versión del SO ya no se encuentra soportada la versión de los paquetes que mantiene no se encuentran actualizados, por lo que las versiones de las herramientas que se instalen pueden que no tengan todas las funcionalidades requeridas, o fallos de seguridad arreglados en futuras versiones.

3.2.3. Otros sistemas

Además de lo anteriormente descrito cabe destacar algunos elementos más que entran en juego a la hora de la comunicación y/o administración del sistema actual.

- **Cortafuegos perimetral**, actualmente un cortafuegos Juniper hace de puerta de acceso a la máquina, dando algo más de seguridad a los servicios de la misma.

- **Armario rack**, como se ha comentado en la subsección 3.2.1 el armario donde actualmente se encuentra el servidor es una parte de suma importancia para el acceso y administración física del mismo. Además todo ello se encuentra en una sala especialmente climatizada para su correcta conservación.

3.3. Necesidades

Las necesidades del sistema se componen de la instalación de una serie de servicios. En estas líneas se detallarán los servicios que se llegarán a instalar, así como unos apuntes a seguir para la consecución del proyecto.

3.3.1. Servicios

- **Servicio de correo electrónico**, el sistema deberá permitir el envío y la recepción de mensajes de correo entre el personal de la empresa, así como la comunicación con otras direcciones externas. El sistema será compatible con los protocolos Simple Mail Transfer Protocol (SMTP), Internet Message Access Protocol (IMAP) y Post Office Protocol (POP). El servidor será compatible tanto con los protocolos cifrados y no cifrados de los mismos, aunque se instará a los usuarios a utilizar el protocolo cifrado. El servidor debe incluir entre sus características el filtrado de correo, así como la detección de correo no deseado. Para ello se instalará el software Antivirus y Antispam necesario. Dicho servidor deberá ser altamente configurable además de estar basado en protocolos estándar para su compatibilidad con la mayoría de los clientes, y de cara a posibles exportaciones o importaciones.

- **Identificación centralizada**, dicha identificación deberá permitir a los usuarios identificarse en sus cuentas utilizando las mismas credenciales utilizadas para el resto de servicios ofrecidos por la empresa. De este modo el usuario no tiene porque recordar un número indeterminado de pares usuario-contraseña. También de este modo se simplifica la administración de los servicios y se minimizan errores.

- **Servicio de agenda y directorio personal**, el sistema contará con una base de datos donde el usuario podrá almacenar y compartir la información referente a su agenda personal, así como un directorio donde almacenar datos de sus contactos. Dichos datos podrán ser accedidos desde cualquier dispositivo conectado a internet.

- **Servicio DNS**, el sistema deberá dar soporte para resolver los nombres y direcciones IP de todos los servicios que alberga, así como nombres relativos al resto de la empresa. También deberá poseer de un servicio de caché de DNS para centralizar todas las peticiones y así minimizar el tráfico DNS hacia internet.

- **Entidad certificadora**, deberá existir una entidad certificadora para la creación y verificación de certificados capaces de cifrar las conexiones de los servicios mediante Secure Socket Layer (SSL). El certificado de dicha entidad deberá estar instalado en todos los equipos que requie-

ran utilizar los servicios por protocolo cifrado, para evitar problemas de confianza entre el software cliente y el servidor.

- **Servicio WEB**, el servicio web deberá dar acceso a las herramientas de correo web para los usuarios, así como ofrecer la posibilidad de la instalación de herramientas de administración vía web para el software que posee el servidor. Para que dichas herramientas puedan ser instaladas será necesario realizar instalaciones de distintos lenguajes de programación así como las librerías correspondientes para su correcto funcionamiento.

- **Servicio de base de datos**, algunos de los datos deben ser almacenados y mantenidos en un Sistema de Gestión de Base de Datos (SGBD). Dicho gestor deberá estar disponible para los distintos servicios, así como para almacenar información de las herramientas de gestión, incluso para almacenar cierta información de los usuarios.

3.3.2. Orientaciones

Los servicios ofrecidos deben estar orientados por las siguientes directrices:

- **Comunicaciones unificadas**, un entorno groupware proveerá de las herramientas necesarias para la comunicación desde todos los miembros de la empresa. Dicha comunicación será independiente de las herramientas utilizadas así como de los dispositivos desde los que se accedan.

- **Alta disponibilidad**, el sistema deberá permanecer disponible el máximo tiempo posible, y el mantenimiento y recuperación ante fallos deberá dejar el sistema caído durante el mínimo tiempo necesario.

- **Administración sencilla**, el sistema proveerá de distintos tipos de administración, para distintos niveles de administrador. Poseerá herramientas web que ofrecerán una interfaz sencilla para los administradores más inexpertos.

- **Conjunto de servicios**, el sistema ofrecerá un conjunto de servicios compuesto por:

  - Servidor de correo electrónico con soporte para los protocolos SMTP, POP y IMAP.

- Servidor de páginas web, para albergar las herramientas de administración, tanto para usuarios como administradores.
- Servidor de agenda y directorio, para almacenar la información personal de los usuarios. Dicha información será almacenada en un SGBD

- **Seguridad y confidencialidad**, el sistema podrá ser accesible mediante protocolos que usen encriptación para una mayor seguridad y confidencialidad de los datos. Por defecto, se instará al usuario a utilizar este tipo de conexiones.

Además hay que tener en cuenta el Presupuesto para la realización del proyecto, que ha de ser lo más bajo posible. Por tanto, se va ha realizar un análisis que determine que se va a ser utilizado para cubrir las necesidades, reduciendo en la medida de lo posible el Presupuesto.

3.4. Solución planteada

En este punto se describe la decisión tomada finalmente para llevar a cabo. Se pretende que la funcionalidad del sistema actual se amplíe, además de introducir nuevas mejoras que eviten los problemas que posee el antiguo sistema así como una mejor administración y mantenimiento del mismo.

Todo el software utilizado será con licencias que permitan su uso sin necesidad de pagar por las mismas.

3.4.1. Sistema Operativo

La elección del sistema operativo viene condicionada por el hardware necesario para cubrir las necesidades del conjunto de SO más los servicios que deber proveer. Al realizar la instalación sobre una máquina virtual, se abstrae todo el compendio de detalles físicos de la máquina atendiendo únicamente a la capacidad de almacenamiento tanto permanente como volátil, capacidad de procesamiento, unidades de red, etc. Es por ello, y partiendo de la base de que ya se dispone de un servidor de máquinas virtuales, los requisitos de la máquina serán los siguientes:

- Sistema con dos procesadores para permitir más ciclos en multi-proceso.

- Capacidad mínima de 1Gb de memoria RAM.
- Al menos 8Gb de disco duro para albergar el sistema y los servicios, y 40Gb para albergar los datos.
- Una interfaz de red para las comunicaciones.

El sistema operativo seleccionado será Ubuntu 12.04 Long Time Support. El cual ofrece un equilibrio entre soporte, ya que según la política del distribuidor dará soporte a los paquetes instalados hasta el año 2017, y las versiones de los programas que albergan son recientes y actualizados. Además el coste de utilización es cero, por lo que se amolda a los requisitos planteados de minimizar el coste en la adquisición de software. Véase la tabla 2.2 en página 38.

3.4.2. Servicios

Los servicios que el sistema necesita están en la subsección 3.3.1. Dichos servicios deben ser suministrados por una serie de programas que están disponibles para el SO que se ha detallado en la subsección 3.4.1. A continuación se especificará uno a uno dichos servicios y los programas elegidos, así como las razones del porque de la elección en concreto.

Servicio de correo electrónico

El sistema requiere de un servidor SMTP, para el envío y recepción de correos, junto a un servidor POP e IMAP que permitan la visualización de los mismos. Dicha configuración permitirá aplicar filtros a los correos, crear carpetas, configurar mensajes de ausencia, etc. Por ello se han elegido los siguientes programas con las configuraciones correspondientes para proveer de un completo sistema de servidor de correo electrónico:

- **Postfix**, este Mail Transport Agent (MTA) es utilizado en múltiples situaciones. Además existe una versión de dicho software ya instalada en la empresa, por lo que se podrán utilizar algunas de sus configuraciones. Por otra parte, es software libre, que es uno de los requisitos indispensables en este proyecto y su actualización y mantenimiento por parte de la comunidad es aceptable en términos de tiempo y fiabilidad. Véase Postfix en página 48.

- **Dovecot**, el servidor de correo no sería utilizable desde dispositivos remotos si no soportase los protocolos IMAP o POP, el software Dovecot proporciona estas funcionalidades, además de acoplarse perfectamente a la filosofía de Postfix y funcionar en conjunto con él. Véase Dovecot en página 45.

- **ClamAV** y **Amavis**, los antivirus de código abierto para servidores de correo electrónico, proporcionarán la seguridad necesaria para filtrar aquellos correos potenciales de ser peligrosos, y así evitar desastres mayores. Véase ClamAV y Amavis en páginas 77 y 78.

- **SpamAssassin**, junto con el antivirus, el software encargado de evitar el correo electrónico no deseado, formarán una plataforma que mantendrá el sistema libre de amenazas. Véase SpamAssassin en página 78.

Autenticación centralizada

- **OpenLDAP**, ofrece las interfaces necesarias para proveer del servicio de directorio que será capaz de mantener la información relativa del usuario, así como podrá almacenar distinta información relativa a la empresa. El sistema OpenLDAP cumple las especificaciones del protocolo Lightweight Directory Access Protocol (LDAP), por lo que ofrece los servicios necesarios para identificar usuarios, devolver información sobre los mismos, etc. Véase OpenLdap en página 53.

Servicio de agenda y directorio personal

- **Horde IMP**, el framework Horde posee multitud de herramientas, todas ellas basadas y distribuidas bajo licencias de software libre, que cumplen cada una de las funciones que aquí se requieren. Los usuarios podrán mantener sus agendas personales sincronizadas entre distintos dispositivos, podrán compartir eventos, o enviar a terceros su disponibilidad, así como mantener su propio directorio de contactos personal. Véase Horde IMP en página 48.

Servicio de DNS

- **DjbDNS**, el servicio DNS será esencial para mantener y enrutar bien todas y cada una de las peticiones realizadas a los distintos sistemas. Así el software proporcionado por *DjbDNS* proporciona las herramientas necesarias para tener un entorno DNS sencillo de mantener y completamente funcional. Véase Djbdns en página 65.

Autoridad de certificación

- **OpenSSL**, el conjunto de herramientas ofrecidas por este software proporcionará la capacidad de crear los certificados necesarios para todos los sistemas, a fin de poseer un entorno que sea capaz de intercambiar información cifrada, además de proporcionar confidencialidad e integridad a las comunicaciones. Véase Openssl en página 67.

Servicio WEB

- **Apache**, será el software encargado de servir las páginas web, y ofrecer la pasarela de acceso al resto de servicios de sincronización. Dicho servidor está ampliamente extendido y la multitud de extensiones que ofrece lo hacen uno de los más recomendables para estas funciones. Véase Apache Web Server en página 58.

- **PHP**, lenguaje utilizado por Horde IMP, por lo que es un requisito indispensable para el correcto funcionamiento de la plataforma. Su instalación junto con el servidor web Apache es uno de los sistemas más utilizados en los servidores actuales. Véase PHP en página 68.

Servicio de base de datos

- **MySQL**, por último, la información del usuario será almacenada en este SGBD. La mayoría del software utilizado en esta plataforma ofrece conectores para los más importantes SGBD pero se ha preferido este sistema dada la experiencia con este tipo de soluciones. Véase Mysql en página 61.

3.5. En este capítulo

En este capítulo se ha dado una visión del sistema actual, de las necesidades que se plantean para sustituirlo, así como las recomendaciones y directrices que se deben tener en cuenta a la hora del diseño.

Además se ha especificado que software cumple con las especificaciones y necesidades detalladas, y es capaz de amoldarse a las recomendaciones realizadas.

En la siguiente parte se dará una visión especifica del problema a resolver, así como la solución planteada que se llevará a implementar más adelante.

4

Diseño

En este capítulo se detallará el diseño final del sistema, así como la comunicación de todas las partes. Se detallarán las bases en cuanto a los sistemas de identificación, seguridad y confidencialidad.

A partir del análisis realizado en el Capítulo 3, se especificarán todos los componentes que conformarán la solución, desde el sistema operativo (SO) hasta los programas necesarios para la sincronización de las agendas.

4.1. Sistema base

El sistema se instalará sobre un SO Ubuntu 12.04 Long Time Support. La instalación se realizará siguiendo las recomendaciones de la página web de la distribución, para dotar al servidor de una mayor seguridad y fiabilidad. Los paquetes a instalar serán siempre descargados de los repositorios oficiales, y se utilizará el comando *apt-get* para su instalación.

La distribución del sistema de ficheros se realizará mediante Logical Volume Manager lo que permitirá una mejor administración del mismo, añadiendo un plus de flexibilidad para poder añadir nuevos volúmenes, extender los ya existentes, o realizar tareas de mantenimiento. Al realizar la instalación sobre sistemas virtuales, la administración hardware del almacenamiento de disco será parte de la administración del hipervisor instalado aunque se recomienda seguir una administración en RAID 51 o RAID 60 para una mayor fiabilidad.

El sistema deberá poseer las herramientas básicas de administración a través de consola remota.

Se recomienda realizar la instalación del servidor Open Secure Shell (OpenSsh) que ayudará a conectarse al servidor manteniendo las comunicaciones cifradas, lo que mantendrá la confidencialidad de las comunicaciones.

Al ser un servidor cuyos servicios son accesibles desde la red, no es necesario poseer un interfaz gráfico que por defecto consumirá recursos innecesariamente. Toda administración mediante herramientas gráficas, debería realizarse desde el ordenador del administrador, para una mejor gestión de los recursos del servidor.

4.2. Comunicaciones

La conexión del sistema con sistemas exteriores se hará mediante una única conexión de red. Sería recomendable mantener una doble conexión, una para los servicios del servidor y otra para la entrada de administración.

Como se puede ver en la figura 4.1, el servidor está tras un cortafuegos que limita las conexiones que se pueden realizar contra el mismo. Al ser un servidor que mantiene los servicios de identificación y se utiliza como entidad de certificación, es recomendable mantenerlo aislado y en una subred propia, protegida mediante cortafuegos, para impedir en la medida de lo posible el acceso al mismo. Al igual que el servidor, la propia red interna deberá estar protegida de cualquier acceso desde internet, bien mediante el router de acceso, que no permita el acceso desde diferentes partes de internet, o bien mediante un cortafuegos posterior, que haga de barrera perimetral.

De esta forma se mantiene una pequeña y primera línea de defensa ante un posible ataque tanto desde el exterior, como desde el interior de la empresa, ya que en el caso de que alguno de los ordenadores interiores se vea comprometido, no sea posible acceder al resto de los servidores del entorno empresarial.

Siempre que fuese posible, se recomienda instalar algunos de los servicios en máquinas separadas. Por mantenimiento de costes, dicha instalación se realizará en una sola máquina, ya que al ser un entorno empresarial pequeño, no requiere de una dedicación exclusiva para cada uno de los servicios, y por lo tanto la instalación se puede realizar en una sola instancia.

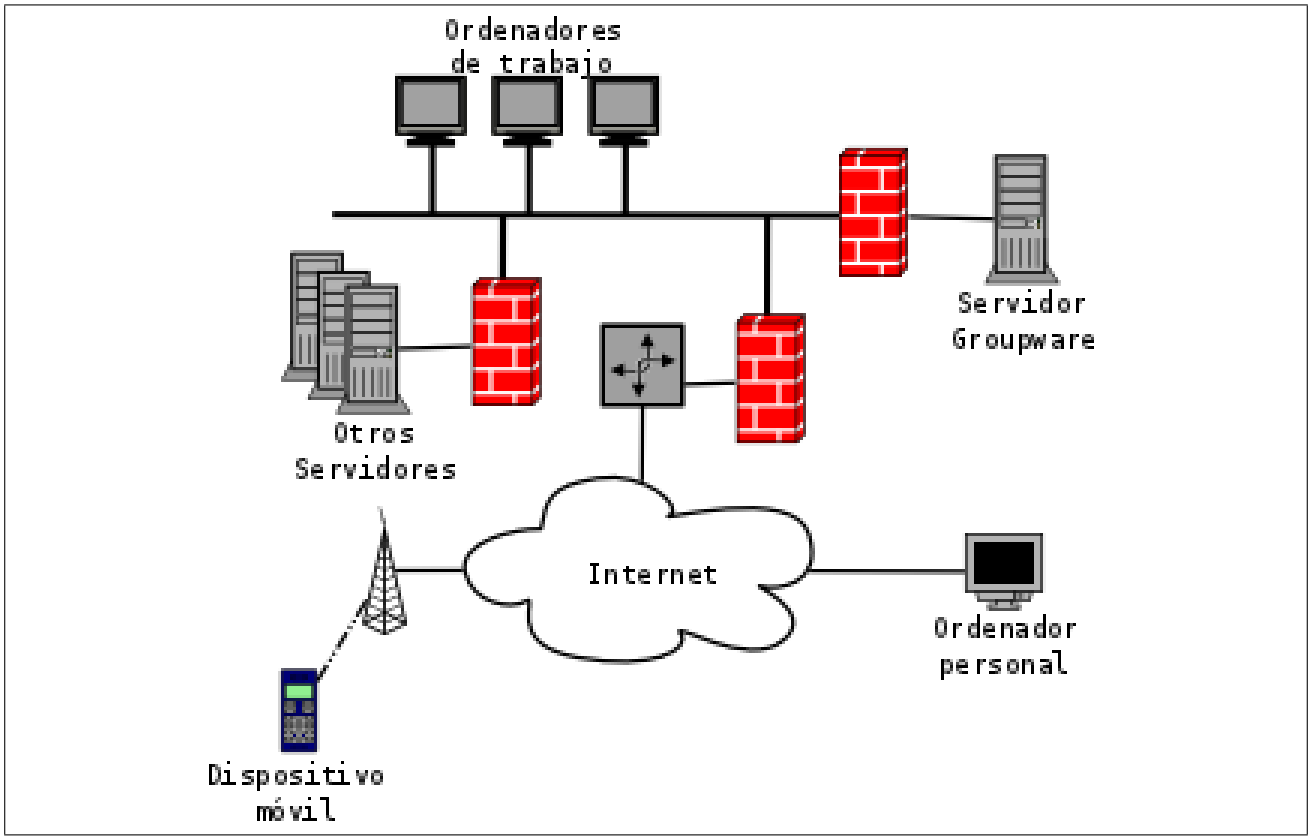

Figura 4.1: *Esquema de conexión general.*

4.3. Servidor de correo

El servidor de correo estará compuesto de los distintos servicios que se especifican a continuación y que se pueden ver en la figura 4.2.

El sentido de las flechas indica la comunicación que se produce entre las distintas partes. Una flecha rellena en su totalidad indica que el acceso es de lectura y escritura, mientras una flecha sin relleno, especifica acceso de sólo lectura y consulta. Los servicios están apoyados en distintos sistemas de almacenamiento, que mantiene la información del usuario, sus credenciales de identificación, los correos, etc. Los servicios que se encuentran dentro de los hexágonos, mantienen una estrecha relación entre ellos, y su conexión se puede dar mediante sockets de Unix, ejecución de ficheros, o comunicación de lectura y escritura entre los sockets TCP abiertos.

- **Servicio Web**: el servicio web será encargado de hacer de pasarela de entrada para las herramientas de groupware. Entre los requisitos que debe cumplir se incluye, soporte y redirección para conexiones mediante Secure Hypertext Transfer Protocol (HTTPS), acceso a todos los servicios de la plataforma groupware. Entre los módulos que debe incluir, se citan, el módulo para conexiones Secure Socket Layer (SSL), soporte para el lenguaje PHP, soporte para memcache e instalación de las dependencias apc, memcache, curl, gd y xml-parse de PHP.

- **Servicio Groupware**: tras la instalación del Horde IMP y la configuración de las entradas correspondientes en el servidor web, el servicio groupware deberá tener las herramientas de correo web, agenda y directorio de contactos entre otras. Además su configuración para su funcionamiento junto a Dovecot debe ser realizada.

- **Servicio MDA**: Encargado de ofrecer los protocolos Post Office Protocol (POP) e Internet Message Access Protocol (IMAP). Se realizará con la instalación de Dovecot y su configuración para funcionar con Postfix.

- **Servicio MTA**: Postfix implementará el servicio MTA, ofreciendo conexión mediante Simple Mail Transfer Protocol (SMTP) para el envío y recepción de correo desde servidores externos.

Todos estos sistemas son accesible desde el exterior, excepto el servicio groupware que se apoya

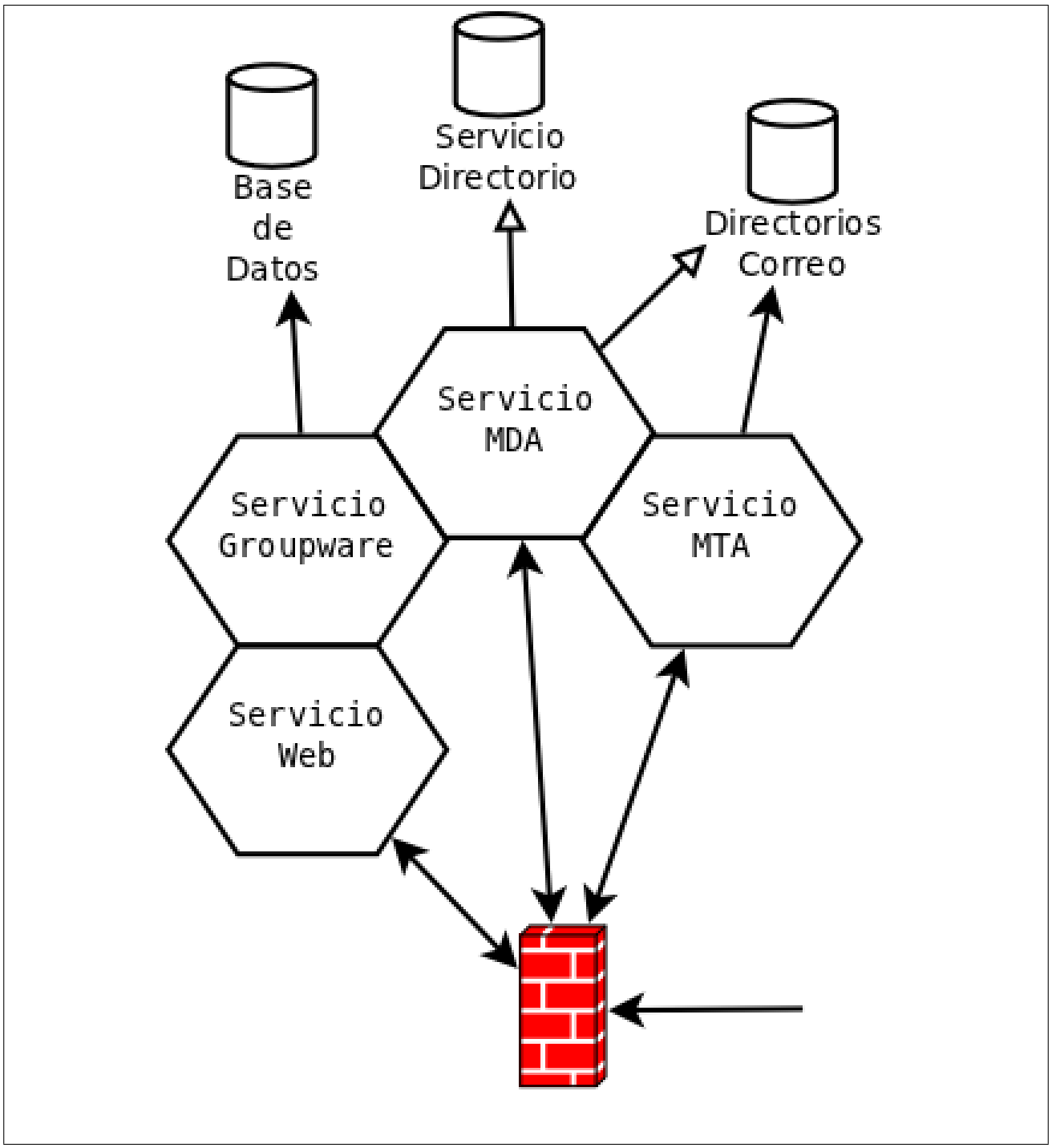

Figura 4.2: *Esquema de conexión entre el servidor de correo y el exterior.*

totalmente en el servidor web, que será el intermediario entre el usuario y los servicios propiamente dichos.

4.4. Seguridad

El sistema de seguridad deberá estar compuesto por distintos dispositivos colocados en distintas posiciones dentro de la red de la empresa, de modo que si un atacante quisiese explotar alguna vulnerabilidad del sistema, no lo encontrase nada fácil.

- **Seguridad perimetral**: Dicha zona de seguridad debe disponerse para cualquier entrada que se realice al sistema desde el exterior de la red interna. Para completar esta zona de seguridad, se dispondrá de un cortafuegos corporativo, encargado de filtrar todas las peticiones que se realicen al servidor, exceptuando aquellas que sean necesarias para la utilización de los servicios de correo, y groupware. Inclusive se recomienda la utilización de un software que realice la tarea de proxy, para así añadir una capa más al diseño permitiendo en un futuro realizar un escalado horizontal de los sistemas.

- **Seguridad interna**: El servidor poseerá una doble barrera de seguridad compuesta por un cortafuegos corporativo externo, más un cortafuegos interno, en este caso Iptables, para impedir cualquier acceso no deseado. Hay que tener en cuenta que direcciones van a realizar el mantenimiento de la máquina, para que dichas direcciones tengan acceso a los servicios de administración.

- **Seguridad de servicios**: Todos los servicios deben ser autenticados, no se podrá realizar ninguna acción sobre los mismos, sin haber realizado previamente la identificación con el servicio correspondiente. Fuera del alcance de la identificación quedan los servicios de consulta de agendas, o recepción de correos.

- **Seguridad de correo**: El sistema poseerá las herramientas pertinentes para asegurar la legitimidad y seguridad de los correos electrónicos recibidos, para ello, deberá disponer de herramientas de Antivirus y Antispam para cumplir este requisito.

4.5. Otros sistemas

El resto de sistemas especificados en la sección de análisis, será instalado siguiendo las recomendaciones del distribuidor del SO. La configuración que se llevará a cabo seguirá las directrices marcadas para asegurar su confidencialidad y mantenimiento. Entre estos sistemas cabe destacar:

- Servidor de identificación basado en Lightweight Directory Access Protocol (LDAP).
- Sistema de Gestión de Base de Datos (SGBD).
- Domain Name System (DNS).
- Certification Authority (CA) basada en Openssl.

El acceso a estos sistemas debe estar protegido mediante pares de usuario y contraseña, además, y siempre que se permita, los sistemas sólo deberán estar escuchando en determinadas direcciones IP, que sólo sean accesibles desde los servicios que lo utilicen, así como desde las herramientas de administración designadas por los administradores. Los accesos no autorizados deben ser notificados y estudiados, para prevenir posibles ataques a la plataforma.

4.6. En este capítulo

Se ha detallado el diseño final del sistema, que comunicación existe entre todas las partes y el detalle último antes de realizar la implementación. Con todos estos datos se procederá a realizar el presupuesto, para continuar con el detalle de los pasos llevados a cabo para la instalación del sistema.

5

Presupuesto

Antes de llevar a cabo la implementación del sistema se detallará el presupuesto y la viabilidad del proyecto. Se mostrarán los gastos calculados, incluyendo tanto directos como indirectos, así como la planificación para su puesta en funcionamiento.

5.1. Actividades

Las actividades que aborda este proyecto están especificadas en la siguiente tabla:

Requisito	Perfil	Jornadas
Análisis y especificación del problema.	Analista Orgánico y Administrador de Sistemas	2
Propuesta y diseño de la solución.	Administrador de Sistemas	3
Ejecución de la solución.	Administrador de Sistemas	10
Documentación.	Jefe de proyecto y Administrador de Sistemas	5
Pruebas.	Administrador de Sistemas	2
Gestión y coordinación del proyecto.	Jefe de proyecto	...
	Total	22

Tabla 5.1: Desglose de actividades.

Las jornadas tendrán una duración de 8 horas, que se realizarán en horario laboral de lunes a

viernes.

La gestión y coordinación del proyecto, será llevada a cabo durante todo el ciclo del proyecto, y se incluirá en el desglose del presupuesto del personal.

5.2. Presupuesto del personal

El precio total del coste de personal se calculará teniendo en cuenta los siguientes puntos:

- Las tareas llevadas a cabo por dos o más perfiles se presupuestará con la mitad de las jornadas a cada uno de los perfiles
- Para la gestión y coordinación del proyecto, se imputan un 10 % de las jornadas totales, en este caso, de un total de 22 jornadas, se imputan 2,2 jornadas para la gestión y coordinación del proyecto.

Por lo tanto la tabla 5.2 muestra la distribución del gasto de personal, y el total presupuestado.

Perfil	Precio/Jornada	Núm. jornadas	Precio
Analista Orgánico	350,00 €	1	350,00€
Administrador de Sistemas	400,00 €	18,5	7.400,00€
Jefe de proyecto	450,00 €	4,7	2.115,00€
	Total	24,2	9.865,00€

Tabla 5.2: Presupuesto económico del personal.

5.3. Distribución temporal

La duración del proyecto según la tabla 5.1 se distribuye en 22 jornadas. La tabla 5.3 muestra como las distintas fases pueden distribuirse de manera que la duración final se amolde a 20 jornadas, o lo que es lo mismo, el proyecto pueda realizarse en 4 semanas laborables.

Semana / Requisito	1	2	3	4
Análisis del problema.				
Diseño de la solución.				
Ejecución de la solución.				
Documentación.				
Pruebas.				
Gestión del proyecto.				

Tabla 5.3: Desglose en el tiempo.

5.4. Recursos materiales

Los recursos materiales empleados en el proyecto incluyen todo el material necesario para la implantación del proyecto. Los principales recursos se componen por la amortización del hardware disponible dentro de la empresa, y que reduciría drásticamente los costes en material. Es importante destacar que el coste de licencias de software es cero, ya que se utilizará software libre para reducir lo máximo posible el coste del mismo.

Recurso	Cantidad	Coste Total
Máquina hardware	1	25,00 €
Máquina hardware	0.50	25,00 €
Máquina hardware	0.50	50,00 €
Máquina hardware	0.50	41,67 €
Switch	0.10	0.83 €
	Total	92,50 €

Tabla 5.4: Recursos materiales utilizados.

5.5. Gastos indirectos y margen de riesgo

Se incluirá un 10% de los costes totales del proyecto para suplir los gastos indirectos, así como manejar un pequeño margen de riesgo. En el Resumen del presupuesto de detallará a cuanto asciende la cantidad de estos gastos.

5.6. Resumen del presupuesto

Concepto	Coste
Gasto personal	9.865,00€
Gasto material	93€
Gastos indirectos	996€
Total	10.953€

Tabla 5.5: Resumen presupuesto.

El presupuesto económico asociado al presupuesto está descrito en la tabla 5.5 y asciende a la cantidad de **10.953€(diez mil novecientos cincuenta y tres euros)**. Este presupuesto no incluye el importe del IVA correspondiente.

Leganés a 28 de Junio de 2013

El Ingeniero proyectista

Fdo. Sergio Montoiro Peinado

UNIVERSIDAD CARLOS III DE MADRID

Escuela Politécnica Superior

PRESUPUESTO DE PROYECTO

1.- Autor: Sergio Montoiro Peinado

2.- Departamento:

3.- Descripción del Proyecto:

- Titulo: **INSTALACIÓN DE UN SERVIDOR GROUPWARE**
- Duración (meses): **1**
- Tasa de costes Indirectos: **10%**

4.- Presupuesto total del Proyecto (valores en Euros):

10.953 Euros

5.- Desglose presupuestario (costes directos)

PERSONAL

Apellidos y nombre	N.I.F. (no rellenar - solo a titulo informativo)	Categoría	Dedicación (hombres mes) [a]	Coste hombre mes	Coste (Euro)	Firma de conformidad
Persona Número 1		Ingeniero Senior	0,925	8.000,00	7.400,00	
Persona Número 2		Ingeniero	0,05	7.000,00	350,00	
Persona Número 3		Ingeniero Senior JP	0,235	9.000,00	2.115,00	
		Hombres mes	**1,21**	**Total**	9.865,00	

[a] 1 Hombre mes = 131,25 horas. Máximo anual de dedicación de 12 hombres mes (1575 horas)

Máximo anual para PDI de la Universidad Carlos III de Madrid de 8,8 hombres mes (1.155 horas)

EQUIPOS

Descripción	Coste (Euro)	% Uso dedicado proyecto	Dedicación (meses)	Periodo de depreciación	Coste imputable [d]
Portátil Trabajo	1.200,00	50	1	24	25,00
Portátil Trabajo	1.200,00	50	1	24	25,00
Portátil Trabajo	1.200,00	100	1	24	50,00
Servidor VMWare	5.000,00	50	1	60	41,67
Switch	500,00	10	1	60	0,83
				Total	92,50

[d] Fórmula de cálculo de la Amortización:

$$\frac{A}{B} \text{xCxD}$$

A = nº de meses desde la fecha de facturación en que el equipo es utilizado
B = periodo de depreciación (60 meses)
C = coste del equipo (sin IVA)
D = % del uso que se dedica al proyecto (habitualmente 100%)

SUBCONTRATACIÓN DE TAREAS

Descripción	Empresa	Coste imputable
	Total	0,00

OTROS COSTES DIRECTOS DEL PROYECTO[e]

Descripción	Empresa	Costes imputable
Licencias Software		0,00
	Total	0,00

[e] Este capítulo de gastos incluye todos los gastos no contemplados en los conceptos anteriores, por ejemplo: fungible, viajes y dietas, otros,...

6.- Resumen de costes

Presupuesto Costes Totales	**Presupuesto Costes Totales**
Personal	9.865
Amortización	93
Subcontratación de tareas	0
Costes de funcionamiento	0
Costes Indirectos	996
Total	10.953

6

Implantación

En este capítulo se especificará la implantación del sistema operativo (SO) y los paquetes de software necesarios, para una puesta en marcha desde cero. A continuación se especificarán los prerrequisitos necesarios, la instalación de los servicios, y se finalizará con la configuración y puesta en marcha del sistema.

6.1. Instalación del SO

Para la instalación se ha seleccionado la siguiente versión de SO: Ubuntu 12.04 Long Time Support. Para descargar la distribución utilizada en este proyecto bastará con ir a la página `http://releases.ubuntu.com/precise/` y descargar la versión que mejor se adapte con nuestro hardware, en este momento existen versiones para procesadores Intel o AMD de 32 o 64 bits.

Una ver arrancada la máquina con el CD o USB de instalación insertado, aparecerá la pantalla(figura 6.1) que nos indicará que el SO está listo para iniciar la instalación.

Para esta guía se utilizará el idioma español, se seleccionará la entrada correspondiente del diálogo de Grub, y comenzará el programa de instalación del SO. El instalador solicitará los datos sobre la disposición del teclado a utilizar(figura 6.2). Al utilizar un teclado estándar español se seleccionará las opciones por defecto. En caso de disponer de otro tipo de teclado, seleccionar el que mejor se adapte al hardware del que dispone la máquina.

A continuación el instalador comprobará si dispone de los controladores necesarios para la insta-

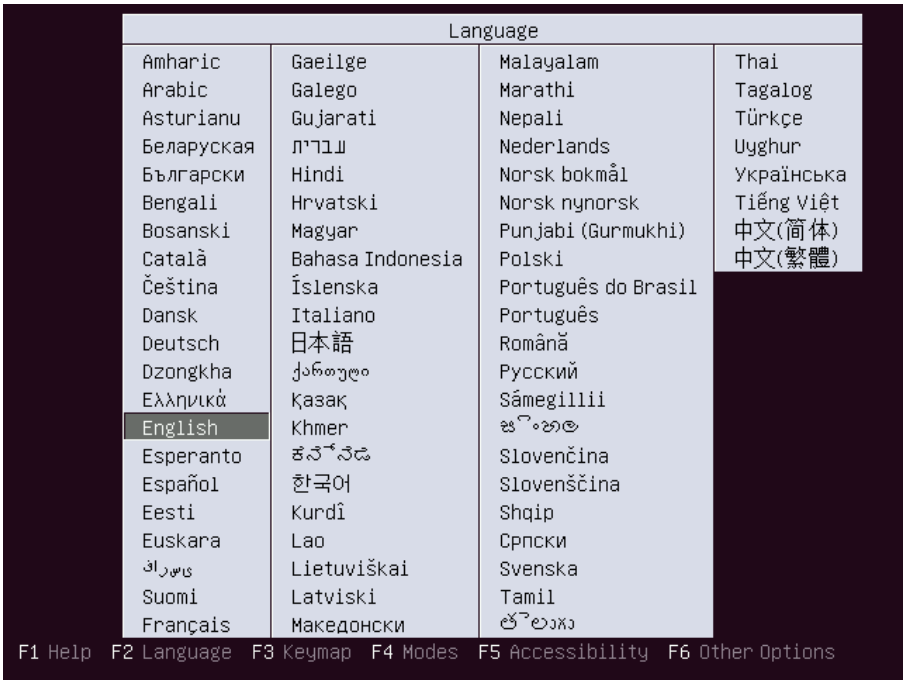

Figura 6.1: *Pantalla de inicio de la carga del CD de instalación.*

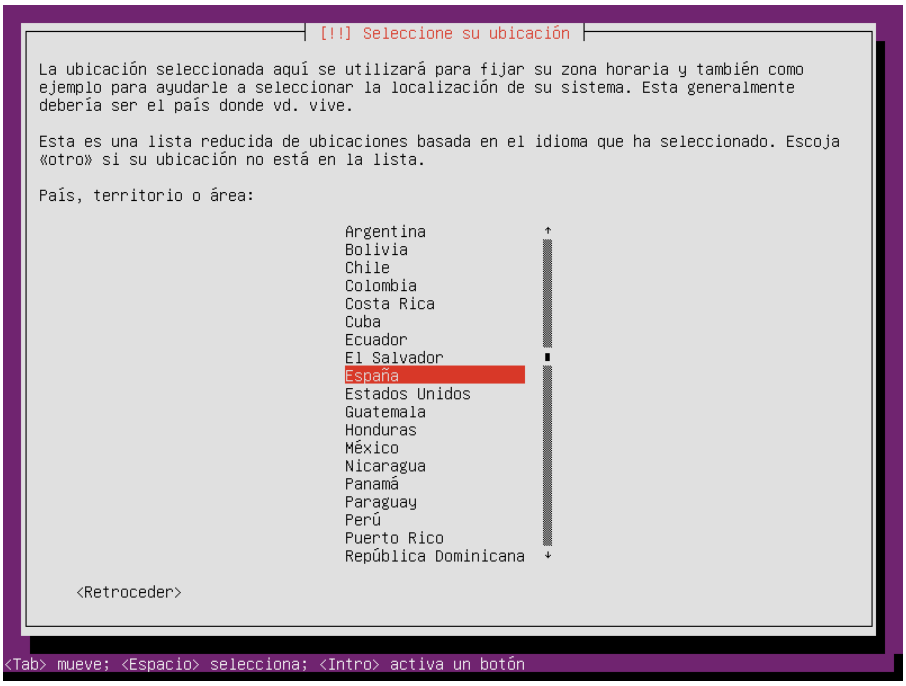

Figura 6.2: *Pantalla de inicio de detección de disposición del teclado.*

lación de los drivers correspondientes, al ser una máquina virtual sobre VMWare no existe ninguna incompatibilidad por lo que se podrá realizar una instalación completa. En caso de tener alguna incompatibilidad de hardware, y en caso de si la instalación no pudiese continuar se solicitarán los drivers adecuados para reanudar con la misma.

El siguiente punto a configurar será de extrema importancia, ya que si en este punto se comete algún error o no se eligen los parámetros correctos, la reconfiguración puede traer algún que otro problema. La configuración de la red requiere de disponer de los datos precisos como son el nombre de la máquina, una dirección IP fija, los datos del Domain Name System (DNS), y otros parámetros que se verán más adelante.

Para este manual en concreto, se eligen los siguientes valores para:

- Nombre: *mail*
- Dominio: *example.com*

El resto de configuración de red dependerá del entorno en el que se vaya a realizar la instalación. Las dirección IP, así como la configuración de puerta de enlace y servidor de nombres de dominio será totalmente diferente en cada una de las máquinas configuradas.

Para el nombre de usuario(figura 6.3) se ha elegido *mailuser* con la contraseña *mailpass*. Al tener un sistema de autorizaciones basado en *sudo*, el instalador otorga a este usuario los permisos necesarios para poder realizar operaciones con privilegios, por lo que es muy importante elegir una buena contraseña y recordarla.

La instalación de la zona horaria también cobra cierta importancia en un servidor de estas características, por lo que seleccionar la zona horaria correcta es fundamental. Durante la instalación(figura 6.4), el programa solicita confirmación sobre la zona horaria detectada, en caso de no ser correcta se mostrará un listado con las distintas zonas, y se deberá seleccionar la zona adecuada.

Una vez realizados estos pasos el instalador intentará recuperar el almacenamiento disponible para la instalación del SO. En caso de no detectar almacenamiento, por ejemplo, el instalador no posee los drivers de la controladora de RAID, el instalador solicitará los drivers necesarios para poder continuar con la operación. Ya que el fin de esta guía no es una configuración hasta el mínimo detalle de la instalación del SO, se dejará por defecto que el instalador elija el particionado utilizan-

Figura 6.3: *Pantalla que solicita el nombre de usuario.*

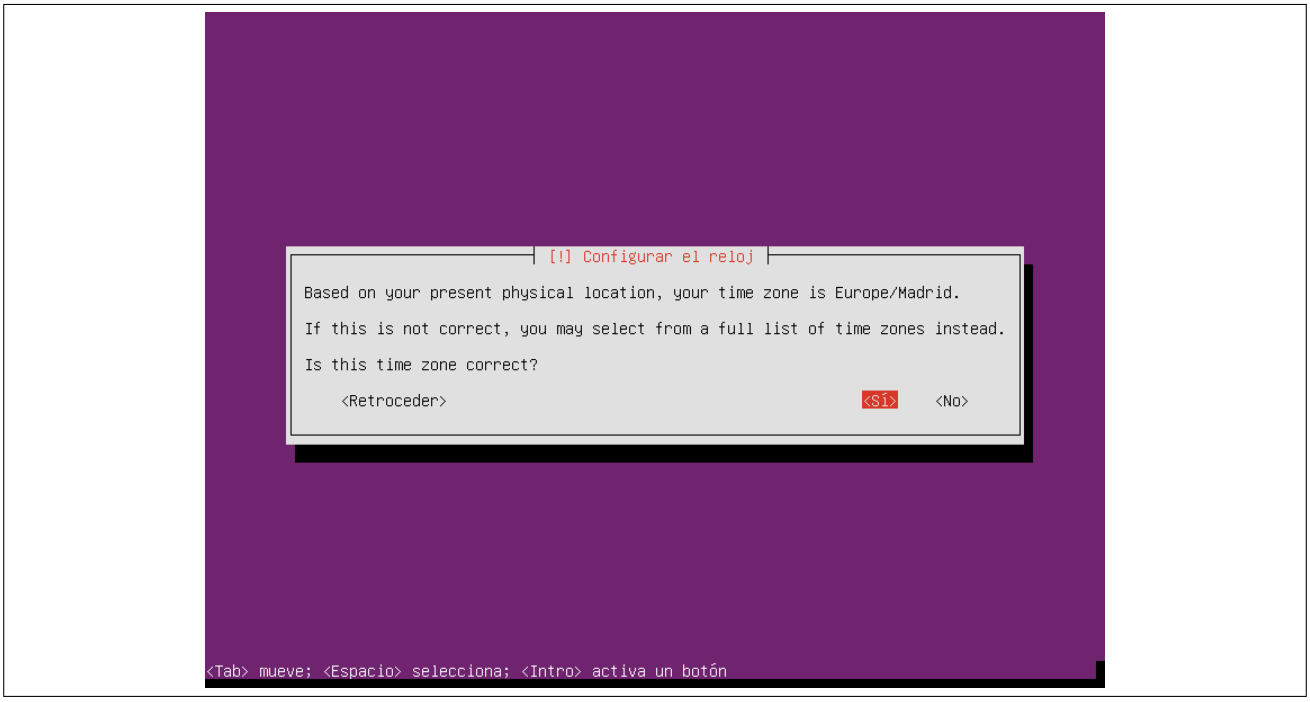

Figura 6.4: *Pantalla de confirmación de selección horaria.*

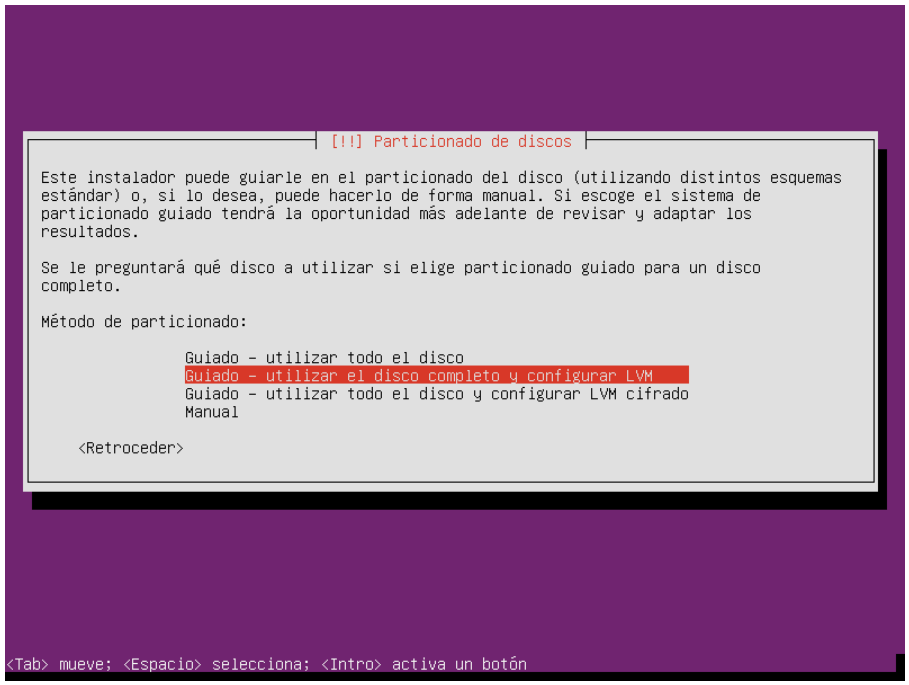

Figura 6.5: *Pantalla de selección de tipo de particionado.*

do Logical Volume Manager(figura 6.5 yfigura 6.6) [22] se selecciona el particionado guiado y se confirman los cambios que realizará el instalador.

La instalación del SO está próxima a su fin. Una ver realizados todos los pasos anteriores, el instalador comenzará a escribir en el disco duro la estructura de ficheros necesaria para funcionar, e instalará los paquetes necesarios para el funcionamiento base, durante esta parte de la instalación tan sólo veremos una barra de progreso que indicará el porcentaje de la instalación de los paquetes. Si el sistema contase con conexión de red, el instalador solicitará configuración de proxy(figura 6.7) para conectarse a los repositorios de software y así descargar y/o actualizar la copia local de los mismos. En caso de que la red no requiera de proxy, bastará con dejar este parámetro en blanco.

El SO Ubuntu permite la configuración automática de actualizaciones críticas o de seguridad en su SO. Esto no es muy recomendable tratándose de sistemas de producción, ya que una actualización de un paquete sin supervisión podría dejar los servicios inaccesibles, por lo tanto, no se configurarán las actualizaciones automáticas(figura 6.8), dichas actualizaciones deberían realizarse bajo supervisión e inclusive sobre sistemas paralelos no críticos, sobre los que se compruebe que las actualizaciones son completamente compatibles con la configuración del sistema que se está insta-

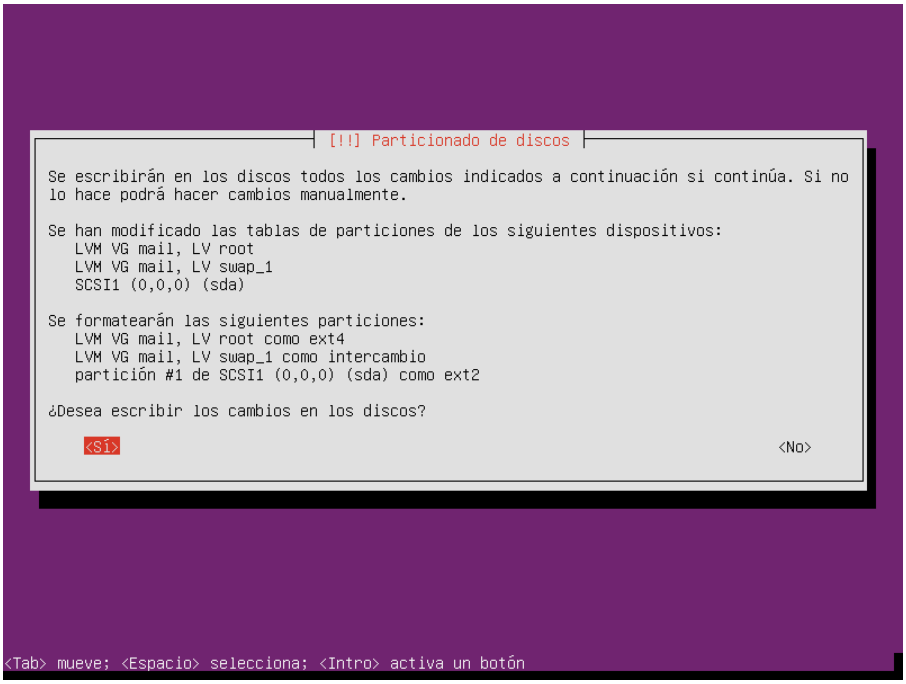

Figura 6.6: *Pantalla de confirmación de particionado y formateo del sistema de ficheros.*

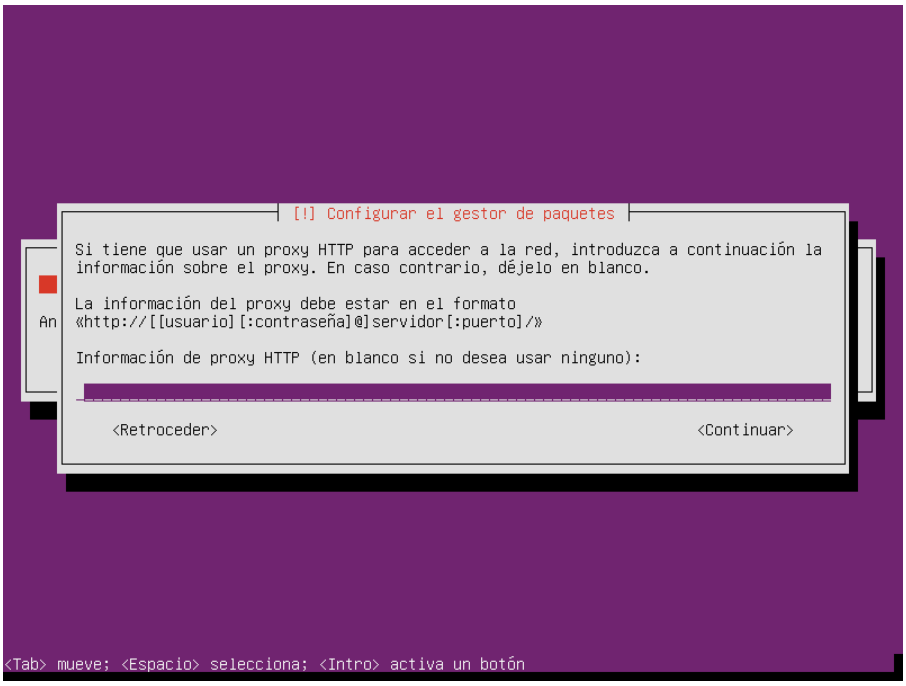

Figura 6.7: *Pantalla de solicitud de configuración de proxy para conexión a internet.*

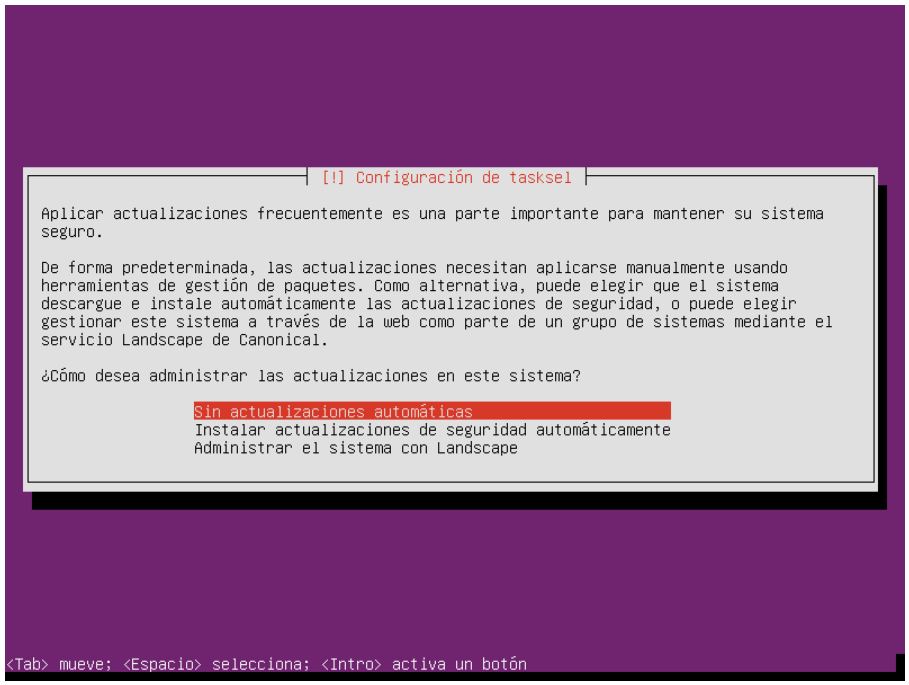

Figura 6.8: *Pantalla de configuración de actualizaciones automáticas.*

lando.

No será necesario la instalación de ningún servicio de los que ofrece el instalador(figura 6.9) a excepción del servidor Open Secure Shell (OpenSsh) que será el que nos deje accesible la máquina a través de una conexión segura remota.

Próximos a la finalización de la instalación, el programa solicitará permisos para instalar el cargador de arranque GRUB. Si en el hardware en el que se está realizando la instalación no se detecta ningún otro SO, el instalador mostrará una pantalla de información(figura 6.10) solicitando confirmación del usuario para instalar el cargador de arranque.

Por último, y una vez instalado por completo el gestor de arranque, se mostrará al usuario la pantalla en la que se solicita la expulsión del CD de instalación para poder reiniciar la máquina, finalizando así la instalación y dando paso a la configuración del SO.

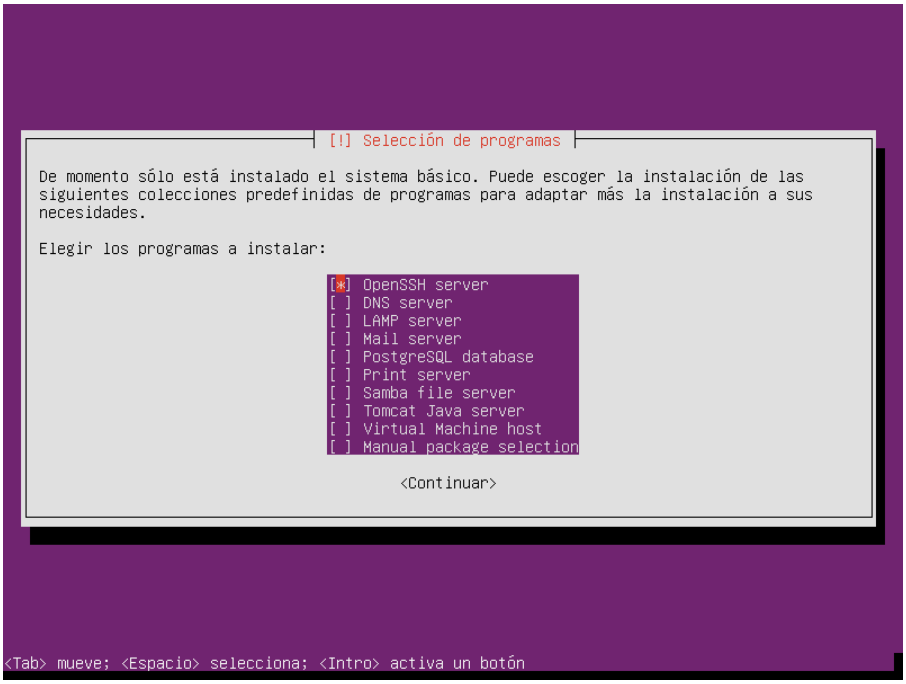

Figura 6.9: *Pantalla para la instalación de servicios sobre el SO.*

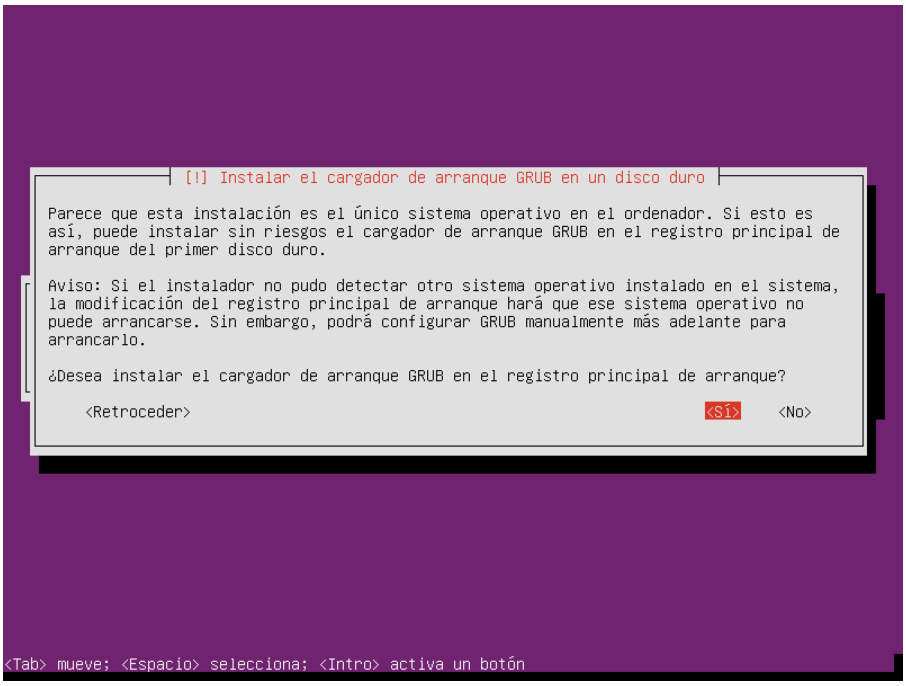

Figura 6.10: *Pantalla de solicitud de confirmación para la instalación de GRUB.*

6.2. Configuración del sistema operativo

Dentro del sistema operativo se realizarán una serie de optimizaciones que se llevarán a cabo para una mejor administración, fiabilidad, y que evitarán posibles ataques a la máquina.

6.2.1. Primeros pasos

Al iniciar por primera vez la máquina hay que comprobar que la configuración realizada durante la instalación es la correcta, para ello habrá que realizar una serie de pequeñas comprobaciones para corroborar la correcta instalación:

- Nombre de la máquina: para comprobar si el nombre de la máquina es correcto, habrá que ejecutar el comando *hostname*. El comando mostrará el nombre que hayamos indicado en la instalación, en este caso *mail*. En el caso que se desee comprobar el nombre completo, junto con el dominio, habrá que añadir al anterior comando el parámetro *-f*.

Código 6.1: Comprobar el nombre de la máquina

```
mailuser@mail:~# hostname
mail
mailuser@mail:~# hostname -f
mail.example.com
```

- Red: el siguiente paso será comprobar la dirección de red, para ello habrá que editar el archivo que se encuentra en */etc/network/interfaces*. Con el editor de texto con el que mas afinidad se tenga, y siempre con permisos de superusuario se modificará el fichero para que tenga un aspecto parecido a este, hay que tener en cuenta que los valores finales dependerán de la configuración de la red donde esté conectado el equipo:

Código 6.2: Cambiar la configuración de red

```
mailuser@mail:~# sudo vi /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static

        address 192.168.82.11 #Dirección ip estática
        netmask 255.255.255.0 #Máscara de red
        network 192.168.82.0 #Red a la que pertenece la máquina
        broadcast 192.168.82.255 #Dirección de broadcast
        gateway 192.168.82.1 #Puerta de enlace por defecto

        dns-nameservers 192.168.82.1 8.8.8.8 #Servidores DNS
```

Una vez que se ha modificado el fichero, o se ha comprobado que su contenido es el correcto, para que el SO recoja los cambios bastará con ejecutar los siguientes comandos:

Código 6.3: Reiniciar la red

```
mailuser@mail:~# sudo ifdown eth0 && sudo iface eth0
```

Para comprobar que los cambios se han realizado con éxito, se comprobará con el comando *ifconfig* que los datos son los correctos.

Código 6.4: Comprobar la configuración de red

```
mailuser@mail:~# ifconfig
eth0      Link encap:Ethernet  direcciónHW 08:00:27:2f:d9:d7
          Direc. inet:192.168.82.11  Difus.:192.168.82.255
Másc:255.255.255.0
          Dirección inet6: fe80::a00:27ff:fe2f:d9d7/64 Alcance:Enlace
          ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST  MTU:1500  Métrica:1
          Paquetes RX:173 errores:0 perdidos:0 overruns:0 frame:0
          Paquetes TX:145 errores:0 perdidos:0 overruns:0 carrier:0
          colisiones:0 long.colaTX:1000
          Bytes RX:22924 (22.9 KB)  TX bytes:16431 (16.4 KB)

lo        Link encap:Bucle local
          Direc. inet:127.0.0.1  Másc:255.0.0.0
          Dirección inet6: ::1/128 Alcance:Anfitrión
          ACTIVO BUCLE FUNCIONANDO  MTU:16436  Métrica:1
          Paquetes RX:96 errores:0 perdidos:0 overruns:0 frame:0
          Paquetes TX:96 errores:0 perdidos:0 overruns:0 carrier:0
          colisiones:0 long.colaTX:0
          Bytes RX:7296 (7.2 KB)  TX bytes:7296 (7.2 KB)
```

- OpenSsh: una vez que la red está configurada y correctamente funcionando el siguiente paso servirá para comprobar si el servidor de consola remoto OpenSsh funciona correctamente. Para ello, y desde una máquina remota y que tenga acceso por red al puerto por defecto de ssh(22) se deberá ejecutar:

Código 6.5: Conexión por ssh

```
ssh mailuser@192.168.82.11
```

La primera vez que se realice la conexión se preguntará si establecer una relación de confianza entre el host local y el remoto. Además cada vez que nos conectemos por ssh a la máquina se presentará un pequeño resumen con los procesos levantados, la carga del sistema, y la utilización tanto de disco como de memoria.

Código 6.6: Salida de primera conexión por ssh

```
The authenticity of host '192.168.82.11 (192.168.82.11)' can't be
established.
ECDSA key fingerprint is
88:5d:e3:d3:6f:e8:f9:8a:75:a6:d8:47:61:b2:4a:1c.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.82.11' (ECDSA) to the list of
known hosts.
mailuser@192.168.82.11's password:
Welcome to Ubuntu 12.04.1 LTS (GNU/Linux 3.2.0-29-generic-pae i686)

 * Documentation:  https://help.ubuntu.com/

  System information as of Sun Nov 25 11:37:23 CET 2012

  System load:  0.28              Processes:           66
  Usage of /:   12.2\% of 7.21GB   Users logged in:     1
  Memory usage: 7\%                IP address for eth0: 192.168.82.11
  Swap usage:   0\%

  Graph this data and manage this system at
        https://landscape.canonical.com/

38 packages can be updated.
23 updates are security updates.

Last login: Thu Oct 11 22:48:45 2012
mailuser@mail:~#
```

- Actualizaciones: antes de comenzar a instalar la lista de paquetes necesarios, es recomendable realizar una actualización completa de los paquetes instalados por defecto, para ello tan sólo habrá que ejecutar el siguiente comando:

Código 6.7: Actualización de los paquetes del sistema

```
mailuser@mail:~# sudo apt-get update && sudo apt-get upgrade
```

En caso de ser necesario se reiniciará el sistema.

Estos pasos iniciales consolidarán la instalación del sistema a su versión más estable, teniendo las últimas actualizaciones que conllevarán una mejora en la seguridad así como un mejor rendimiento del mismo.

6.2.2. Seguridad

Los primeros pasos para asegurar el equipo contra ataques estarán basados en una correcta actualización a la última versión conocida de los paquetes, actualización que se ha llevado a cabo en el último punto del apartado anterior(Código 6.7), y una configuración básica para asegurar que sólo se pueda acceder al equipo mediante consola remota, o ssh, a través de una serie de direcciones predefinidas en los ficheros */etc/hosts.allow* y */etc/hosts.deny*.

Dichos ficheros permiten al TCP Wrapper aceptar o denegar la utilización de la red por parte de los servicios instalados. De esta forma se consigue que los servicios esenciales puedan ser sólo accedidos desde direcciones conocidas.

Código 6.8: Contenidos ficheros hosts.allow hosts.deny para sshd

```
mailuser@mail:~# sudo cat /etc/hosts.allow
...
sshd: 192.168.82.0/255.255.255.0
...

mailuser@mail:~# sudo cat /etc/hosts.deny
...
sshd: ALL
...
```

También será necesaria la configuración de algunas reglas a nivel de IPTables para asegurar un mayor control de que accesos tiene la máquina y que intentos o ataques pueda sufrir, a parte de limitar también las conexiones a nivel de kernel [14]. Antes de comenzar con la configuración para

ssh se deberán incluir unas reglas básicas para permitir las conexiones en la interfaz de red *lo*, las establecidas, así como las conexiones salientes [3].

Código 6.9: Configuración iptables por defecto

```
mailuser@mail:~# sudo iptables -A INPUT -i lo -j ACCEPT
mailuser@mail:~# sudo iptables -A INPUT -m state --state\
 ESTABLISHED,RELATED -j ACCEPT
mailuser@mail:~# sudo iptables -P OUTPUT ACCEPT
```

En este punto hay que tener mucho cuidado, ya que una mala configuración de las reglas puede dejar el sistema sin comunicación, y haciendo que la configuración y/o reseteo de sucesivas reglas se deba ejecutar desde la consola de la máquina en local.

Código 6.10: Configuración iptables para ssh

```
mailuser@mail:~# sudo iptables -A INPUT -p tcp -m state --state NEW\
 --source 192.168.82.0/24 --dport 22 -j ACCEPT
mailuser@mail:~# sudo iptables -A INPUT -p tcp --dport 22 -j DROP
mailuser@mail:~# sudo iptables -A INPUT -p tcp --dport 22 -m limit \
--limit 5/min -j LOG --log-prefix "Denied␣SSH:␣" --log-level 7
```

De esta forma insertamos las reglas que permiten el acceso al puerto de OpenSsh sólo desde la subred del equipo y bloqueamos el resto de conexiones. Durante todo el documento se irán añadiendo nuevas reglas a las anteriormente descritas par así permitir conectar con el servidor en los puertos o servicios especificados. Además se incluirán una serie de reglas que bloquearán e introducirán una entrada en el log de todos aquellos paquetes que no satisfagan ninguna de las reglas anteriormente descritas.

Código 6.11: Entrada para escribir en el log las conexiones no permitidas

```
mailuser@mail:~# sudo iptables -A INPUT -p tcp -j DROP
mailuser@mail:~# sudo iptables -A INPUT -p tcp -m limit --limit 5/min\
 -j LOG --log-prefix "Denied␣TCP:␣" --log-level 7
mailuser@mail:~# sudo iptables -A INPUT -p udp -j DROP
mailuser@mail:~# sudo iptables -A INPUT -p udp -m limit --limit 5/min\
 -j LOG --log-prefix "Denied␣UDP:␣" --log-level 7
mailuser@mail:~# sudo iptables -A INPUT -p icmp -j DROP
mailuser@mail:~# sudo iptables -A INPUT -p icmp -m limit --limit 5/min\
 -j LOG --log-prefix "Denied␣ICMP:␣" --log-level 7
```

Una vez realizados estos cambios, para comprobar que todos se hayan guardado con éxito bastará con ejecutar el comando:

Código 6.12: Comando para comprobar el estado de iptables

```
mailuser@mail:~# sudo iptables -L -v
```

La salida nos dará las reglas que están aplicando en este momento. Además para hacer que las reglas sean persistentes durante los reinicios habrá que incluir un par de scripts en los directorios */etc/network/if-pre-up.d* y */etc/network/if-post-down.d* de esta forma cada vez que la interfaz de red cambie de estado, ejecutará los scripts que habilitan IPTables o guardan las reglas en el fichero */etc/iptables.rules*.

Código 6.13: Contenido de los ficheros iptablesload e iptablessave

```
mailuser@mail:~# sudo cat /etc/network/if-pre-up.d/iptablesload
#!/bin/sh
iptables-restore < /etc/iptables.rules
exit 0

mailuser@mail:~# sudo cat /etc/network/if-post-down.d/iptablessave
#!/bin/sh
iptables-save -c > /etc/iptables.rules
if [ -f /etc/iptables.downrules ]; then
   iptables-restore < /etc/iptables.downrules
fi
exit 0
```

Dichos comandos guardan la configuración o la restauran a la última guardada. Se puede encontrar el fichero con todas las reglas de IPTables en */etc/iptables.rules*.

6.2.3. Otros ajustes

Además de las configuraciones anteriores, también serán necesarios otra serie de ajustes para un correcto funcionamiento.

- Configuración del reloj: Para tener el reloj correctamente configurado y sincronizado, además de seleccionar la zona horaria correspondiente, no está de más instalar el programa *ntpdate* para realizar sincronizaciones con otros relojes diariamente.

  Código 6.14: Instalación del paquete ntpdate

  ```
  mailuser@mail:~# sudo apt-get install ntp ntpdate
  ```

  Después de la instalación, y para asegurar una correcta sincronización, se recomienda crear un archivo ejecutable en las tareas del cron para ejecutarse una vez al día. Para ello se creará el archivo */etc/cron.daily/ntpdate* con el siguiente contenido:

Código 6.15: Contenido del fichero /etc/cron.daily/ntpdate

```
ntpdate ntp.ubuntu.com
```

No hay que olvidarse de dar los permisos de ejecución adecuados al fichero con el comando *chmod*

Código 6.16: Cambio de permisos sobre el fichero /etc/cron.daily/ntpdate

```
mailuser@mail:~# sudo chmod 755 /etc/cron.daily/ntpdate
```

6.3. Prerrequisitos

El resto de la instalación de las distintas herramientas, requiere de la instalación y configuración de algunos paquetes. En las siguientes secciones se especificarán los pasos y paquetes previos a la instalación de todos y cada uno de los distintos componentes del servidor final.

6.3.1. Openssl

Para una mayor seguridad y confidencialidad de los datos intercambiados entre el servidor y los clientes será necesario que los distintos servicios manejen sus comunicaciones de forma cifrada. Algunos como OpenSsh utilizan un protocolo encriptado y difícilmente franqueable por un atacante. Otros como Hypertext Transfer Protocol (HTTP) necesitan de una extensión basada en Secure Socket Layer (SSL) para un correcto funcionamiento en comunicaciones cifradas. En este capítulo se introducirá la instalación y una configuración mínima y autosuficiente para la correcta comunicación entre el servidor y los clientes [11].

Instalación Openssl

La instalación en Ubuntu de los paquetes necesarios se realizará a través del comando *apt-get*.

Código 6.17: Instalación de los paquetes asociados a ssl

```
mailuser@mail:~# sudo apt-get openssl
```

Entidad Certificadora

En este caso, ya que el servidor y todos los sistemas que se van a conectar al mismo serán accesibles sólo por equipos internos, se podrá crear una autoridad certificadora para la firma y validación de certificados.

Este paso puede omitirse si se dispone en el entorno de trabajo de una entidad certificadora. En dicho caso se podrá crear directamente los certificados del servidor, y posteriormente se enviarán a la entidad correspondiente para su firma. Esto no exime de la instalación del paquete *openssl* que será necesario para el resto de operaciones a realizar.

Para crear la entidad certificadora se utilizará el script *CA.pl* de *openssl* que está ubicado en el directorio */usr/lib/ssl/misc*. Además la instalación de todos los certificados firmados se realizará en el directorio local */etc/ssl/local/* teniendo dentro de dicho directorio todos los elementos necesarios para un correcto funcionamiento, si el directorio no existe habrá que crearlo y darle los permisos correspondientes.

Código 6.18: Creación directorio ssl y permisos asociados

```
mailuser@mail:~# sudo mkdir -p /etc/ssl/local
mailuser@mail:~# sudo chmod 700 /etc/ssl/local
```

Antes de comenzar con la creación de los certificados, sería necesario completar el fichero de configuración por defecto para utilizar en todos ellos, *openssl.cnf*, de esta forma será más sencillo el crear los ficheros posteriores y se evitarán posibles errores a la hora de la creación de los certificados.

Código 6.19: Valores por defecto para todos los ficheros de certificados que serán creados

```
mailuser@mail:~# sudo cat /etc/ssl/openssl.cnf
# OpenSSL fichero de configuracion.
# Directorio de trabajo.

# Variable $dir
dir = /etc/ssl/local

[ ca ]
default_ca = CA_default

[ CA_default ]
serial = $dir/serial
database = $dir/certindex.txt
new_certs_dir = $dir/certs
certificate = $dir/cacert.pem
private_key = $dir/private/cakey.pem

default_days = 365
default_md = md5
preserve = no
email_in_dn = no
nameopt = default_ca
certopt = default_ca
policy = policy_match

[ policy_match ]
countryName = match
stateOrProvinceName = match
organizationName = match
organizationalUnitName = optional
commonName = supplied
emailAddress = optional
```

```
[ policy_anything ]
countryName = optional
stateOrProvinceName = optional
organizationName = optional
organizationalUnitName = optional
commonName = supplied
emailAddress = optional

[ req ]
default_bits = 1024 # Size of keys
default_keyfile = key.pem # name of generated keys
default_md = md5 # message digest algorithm
string_mask = nombstr # permitted characters
distinguished_name = req_distinguished_name
req_extensions = v3_req

[ req_distinguished_name ]
# Nombre de Variable    Diálogo de openssl
0.organizationName = Organization Name (company)
organizationalUnitName = Organizational Unit Name (department, division)
emailAddress = Email Address
emailAddress_max = 40
localityName = Locality Name (city, district)
stateOrProvinceName = State or Province Name (full name)
countryName = Country Name (2 letter code)
countryName_min = 2
countryName_max = 2
commonName = Common Name (hostname, IP, or your name)
commonName_max = 64
```

```
# Valores por defecto, para consistencia y ahorro al teclear.
# Nombre de variable            Valor
0.organizationName_default = Example Com.
localityName_default = Leganes
stateOrProvinceName_default = Madrid
countryName_default = ES
emailAddress_default = smontoiro@example.com

[ v3_ca ]
basicConstraints = CA:TRUE
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always,issuer:always

[ v3_req ]
basicConstraints = CA:FALSE
subjectKeyIdentifier = hash
```

Del anterior fichero tan sólo tener en cuenta los diálogos que va a utilizar Openssl para preguntar por los valores que necesita y los valores que tomará por defecto para algunas de las características, dichos valores serán completados con los valores que se requiera en cada caso. Una vez que dicho fichero existe se ejecutarán los comandos correspondientes del script *CA.pl* para crear toda la estructura necesaria para comenzar a firmar los certificados. Por defecto se utilizarán los valores que proponga Openssl, que serán los almacenados en el fichero *openssl.cnf* que acaba de ser editado. Para que el programa CA.pl funcione de forma correcta con la estructura de directorios propuesta, sería recomendable hacer una modificación para que el directorio de trabajo por defecto fuese */etc/ssl/local*.

Código 6.20: Cambio valor variable $CATOP

```
...
#\$CATOP="./demoCA";
\$CATOP="/etc/ssl/local";
...
```

Además, hay que crear los ficheros *cetindex.txt* y *serial* para un correcto funcionamiento, bastará con ejecutar:

Código 6.21: Configuración del directorio de la entidad certificadora

```
mailuser@mail:~# sudo touch /etc/ssl/local/certindex.txt
mailuser@mail:~# sudo echo '100001' > /etc/ssl/local/serial
```

Una vez ejecutado el programa, la salida tendría que ser parecida al código que se ve a continuación. Notar que habrá que elegir contraseñas seguras tanto para el certificado, como para la clave privada, dichas contraseñas serán necesarias a la hora de firmar certificados, por lo que deben almacenarse en un lugar seguro. Aunque el fichero */etc/ssl/local/cacert.pem* debe ser distribuido entre todos los dispositivos que vayan a tener acceso a dicho servidor y quieran confiar en los certificados emitidos por dicha entidad.

Código 6.22: Creación entidad certificadora

```
mailuser@mail:~# sudo /usr/lib/ssl/misc/CA.pl -newca
CA certificate filename (or enter to create)

Making CA certificate ...
Generating a 1024 bit RSA private key
.........++++++
.........................++++++
writing new private key to '/etc/ssl/local/private/cakey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Organization Name (company) [Example Org.]:
Organizational Unit Name (department, division) []:
Email Address [smontoiro@example.com]:
Locality Name (city, district) [Leganes]:
State or Province Name (full name) [Madrid]:
Country Name (2 letter code) [ES]:
Common Name (hostname, IP, or your name) []:ca.example.com
Using configuration from /usr/lib/ssl/openssl.cnf
Enter pass phrase for /etc/ssl/local/private/cakey.pem:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
organizationName      :PRINTABLE:'Example Org.'
localityName          :PRINTABLE:'Leganes'
stateOrProvinceName   :PRINTABLE:'Madrid'
countryName           :PRINTABLE:'ES'
commonName            :PRINTABLE:'ca.example.com'
Certificate is to be certified until Mar 15 17:01:52 2016 GMT (1095 days)
Write out database with 1 new entries
Data Base Updated
```

Creación de certificados

Para la creación de una clave privada y su correspondiente petición de certificado para los servicios del servidor se crearán una serie de claves que corresponderán a los distintos nombres con los que se denominarán cada uno de los servicios. Se realizará el certificado para el sitio *mail.example.com*, el resto de certificados se realiza del mismo modo, se dispondrá de los certificados para *webmail.example.com* y *smtp.example.com* de esta forma, los distintos nombres y certificados, nos permitirán escalar el sistema horizontalmente. Para la creación del certificado bastará con seguir las siguientes indicaciones.

Código 6.23: Creación certificado mail.example.com

```
mailuser@mail:~# sudo /usr/lib/ssl/misc/CA.pl -newreq-nodes
Generating a 1024 bit RSA private key
........++++++
...........++++++
writing new private key to 'newkey.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Organization Name (company) [Example Org.]:
Organizational Unit Name (department, division) []:
Email Address [smontoiro@example.com]:
Locality Name (city, district) [Leganes]:
State or Province Name (full name) [Madrid]:
Country Name (2 letter code) [ES]:
Common Name (hostname, IP, or your name) []:mail.example.com
Request is in newreq.pem, private key is in newkey.pem
```

Se han creado varios ficheros, que se pueden ver en el propio directorio. Dichos ficheros contienen la clave privada del certificado así como la petición para la firma del mismo. Para firmar dichos ficheros bastará con ejecutar lo siguiente, y una vez ejecutado habrá que mover los ficheros a los

directorios correspondientes para poder utilizarlos por los servicios.

Código 6.24: Listado de ficheros una vez firmado el certificado

```
mailuser@mail:~# sudo /usr/lib/ssl/misc/CA.pl -sign
Using configuration from /usr/lib/ssl/openssl.cnf
Enter pass phrase for /etc/ssl/local/private/cakey.pem:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
organizationName      :PRINTABLE:'Example Org.'
localityName          :PRINTABLE:'Leganes'
stateOrProvinceName   :PRINTABLE:'Madrid'
countryName           :PRINTABLE:'ES'
commonName            :PRINTABLE:'mail.example.com'
Certificate is to be certified until Mar 16 17:23:07 2014 GMT (365 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
Signed certificate is in newcert.pem
```

Código 6.25: Listado de ficheros una vez firmado el certificado

```
mailuser@mail:~# ls -l
total 12
-rw-r--r-- 1 root root 2433 mar 16 18:23 newcert.pem
-rw-r--r-- 1 root root 1041 mar 16 18:15 newkey.pem
-rw-r--r-- 1 root root  777 mar 16 18:15 newreq.pem
```

Código 6.26: Copiado de clave privada y certificado firmado

```
mailuser@mail:~# sudo mv newkey.pem /etc/ssl/private/mail.example.com.key
mailuser@mail:~# sudo mv newcert.pem /etc/ssl/certs/mail.example.com.cert
```

Para el resto de certificados las operaciones a realizar son exactamente las mismas, modificando el *Common Name* que se ha especificado en este último. Dichos certificados pueden ser utilizados

sin ningún problema por cualquier programa que soporte cifrado SSL. Para comprobar que el certificado es válido y tiene todos los datos necesarios, se pueden ejecutar los siguientes comandos.

Código 6.27: Operaciones para comprobar los certificados

```
mailuser@mail:~# sudo openssl x509 -noout -text \
        -in /etc/ssl/certs/mail.example.com.cert
Certificate:
    Data:
        Version: 1 (0x0)
        Serial Number: 1048578 (0x100002)
    Signature Algorithm: md5WithRSAEncryption
        Issuer: C=ES, ST=Madrid, O=Example Org., CN=ca.example.com
        Validity
            Not Before: Mar 16 17:38:44 2013 GMT
            Not After : Mar 16 17:38:44 2014 GMT
        Subject: C=ES, ST=Madrid, O=Example Org., CN=mail.example.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (1024 bit)
                Modulus:
                    00:a7:60:a8:f3:66:8d:5f:80:ee:31:4b:cb:b9:8b:
                    ed:15:d4:b7:47:c6:c0:8c:c1:ee:73:99:ae:50:a0:
                    d2:0c:a9:fe:77:d6:eb:43:d4:df:35:8b:80:a0:15:
                    cb:20:7d:6f:c3:46:62:61:5f:db:c7:c4:69:72:e9:
                    7d:3b:e3:c3:8a:c5:83:42:91:25:3d:48:6c:4f:8a:
                    3d:08:bd:e9:c3:f8:52:5c:9d:43:b8:3e:23:d5:d1:
                    74:58:52:8e:84:10:7f:14:4d:7b:27:40:77:38:e3:
                    98:6c:20:3a:63:b4:55:df:ff:cf:ac:f3:a8:c6:f1:
                    9c:bb:e3:07:08:6f:fa:40:eb
                Exponent: 65537 (0x10001)
    Signature Algorithm: md5WithRSAEncryption
         b7:9a:49:19:67:af:35:33:05:f5:a7:01:2c:c5:24:0d:e2:a0:
         e6:e8:94:83:3d:9b:53:a9:f4:dc:4b:48:d1:e2:a0:12:7e:3d:
         be:f3:df:5a:66:2a:d2:78:a1:04:74:fb:99:e7:ee:6a:05:a8:
         bc:e1:a3:87:95:09:5e:b0:e0:f9:88:01:14:1a:cf:a6:e1:d4:
         98:72:99:01:0f:c7:5c:8f:12:30:77:f3:34:3d:52:70:a1:ff:
         0f:75:77:88:02:7f:a6:45:a5:5f:9e:32:3e:38:a9:24:6a:a5:
         36:6e:27:75:d1:27:37:42:00:7e:e4:42:33:64:52:77:88:58:
         5a:7f
```

6.3.2. Servidor LAMP

Un servidor LAMP [19] comprende la instalación de un SO con un conjunto de programas de software.

- **L**inux: SO utilizado para correr el resto de programas.
- **A**pache: Servidor web de código abierto, uno de los más utilizados [13].
- **M**ySQL: Sistema de Gestión de Base de Datos (SGBD) relacional.
- **P**hp: Lenguaje de programación diseñado para producir sitios web dinámicos.

Es necesaria la instalación de dicho servidor para poder brindar a los usuarios una completa experiencia desde los distintos dispositivos desde los que se pueda conectar, así como ofrecer una interfaz web para tareas de consulta y/o administración.

Instalación servidor LAMP

Al igual que el resto de programas, la instalación en Ubuntu se realizará gracias al comando *apt-get*.

Código 6.28: Instalación de los paquetes asociados al servidor LAMP

```
mailuser@mail:~# sudo apt-get install lamp-server^
```

Es importante resaltar el símbolo ^ insertado al final de la línea, dicho símbolo es necesario para instalar todos y cada uno de los paquetes necesarios para un correcto funcionamiento.

Durante la instalación aparecerá un diálogo en el que se solicitará la contraseña de administración de base de datos. Dicha contraseña debe ser recordara ya que permitirá acceder al gestor para realizar las modificaciones pertinentes.

Para comprobar que la instalación ha finalizado con éxito, bastará con realizar unas pequeñas pruebas sobre el servidor:

- Servidor web: bastará con acceder a la url *http://ip-servidor/* y comprobar que aparece el mensaje de apache *It works*

- Base de datos: desde el mismo equipo y en la consola, se ejecutará el comando *mysql -u root -p*. Aparecerá un diálogo solicitando la contraseña que se ha especificado en la instalación. Si se accede al SGBD, la instalación ha sido satisfactoria.

Código 6.29: Conexión al SGBD

```
mailuser@mail:~# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \\g.
Your MySQL connection id is 41
Server version: 5.5.28-0ubuntu0.12.04.3 (Ubuntu)

Copyright (c) 2000, 2012, Oracle and/or its affiliates.
 All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\\h' for help. Type '\\c' to clear the
 current input statement.

mysql>
```

- Lenguaje de programación PHP: para ello se deberá guardar el siguiente fichero en el directorio */var/www*

Código 6.30: Contenido fichero /var/www/test.php

```
<?php
 phpinfo();
?>
```

Es importante que dicho fichero sea del usuario que ejecute el demonio de apache. En Ubuntu por defecto *www-data*. Para ello bastará con ejecutar:

Código 6.31: Contenido fichero test.php

```
mailuser@mail:~# sudo chown www-data.www-data /var/www/test.php
```

Por último y accediendo desde un explorador web a la dirección *http://ip-servidor/test.php* se obtendrá un resumen de la instalación de php en la máquina.

Otros ajustes

Para un correcto funcionamiento de los servicios que ejecutará este servidor, será necesario instalar y configurar una serie de paquetes:

- **Memcached**: Dicho paquete es necesario para que el servidor soporte las aplicaciones *webmail* que van a ser instaladas. Su instalación es tan sencilla como ejecutar el comando:

  Código 6.32: Instalación del paquete memcached

  ```
  mailuser@mail:~# sudo apt-get install memcached
  ```

  Una vez instalado no es necesaria ninguna configuración adicional, en este momento, para este paquete. En la siguiente sección se detallará cómo el compilador *php* hace uso de este paquete.

- **Dependencias PHP**: Los paquetes listados a continuación, son necesarios para un correcto funcionamiento de la plataforma. Los programas que se van a instalar, dependen de ciertas librerías PHP que no son instaladas por defecto. Dichas librerías comprenden desde el manejo de la cache *APC*, soporte para el paquete *Memcached* instalado en la sección anterior, y otro tipo de funcionalidades.

  Código 6.33: Instalación de dependencias PHP

  ```
  mailuser@mail:~# sudo apt-get install php-apc php5-memcache \
  php5-curl php5-gd php-xml-parser php5-imap
  ```

  La configuración por defecto de PHP se encuentra en el archivo */etc/php5/apache2/php.ini*. No es necesario realizar ningún ajuste extra al compilador, sin embargo, y si se necesitaran realizar algunos, las modificaciones se realizarán en el fichero descrito.

6.3.3. Servidor LDAP

Para comenzar con la instalación del servidor Lightweight Directory Access Protocol (LDAP) bastará con poner en la consola la siguiente instrucción:

Código 6.34: Instalación del servidor ldap

```
mailuser@mail:~# sudo apt-get install slapd ldap-utils
```

Durante las instalación se solicitará a través de un diálogo la contraseña para el usuario administrador del árbol LDAP, dicha contraseña deberá ser almacenada para realizar los cambios correspondientes en el directorio. Una vez realizada la instalación,se realizará la primera configuración. Para ello se siguen las instrucciones que aparecen a continuación.

Código 6.35: Configuración del servidor ldap

```
mailuser@mail:~# sudo apt-get install dpkg-reconfigure slapd

¿Desea omitir la configuración del servidor OpenLDAP? <No>
Introduzca su nombre de dominio DNS: example.com
Nombre de la organización: example.com
Contraseña del administrador: contraseña
Verificación de contraseña: contraseña
Motor de base de datos a utilizar:HDB
¿Desea que se borre la base de datos cuando se purgue el paquete slapd? <No>
¿Desea mover la base de datos antigua? <Sí>
¿Desea permitir el protocolo LDAPv2? <No>

 * Stopping OpenLDAP slapd                                [ OK ]
  Moving old database directory to /var/backups:
  - directory unknown... done.
  Creating initial configuration... done.
  Creating LDAP directory... done.
 * Starting OpenLDAP slapd                                [ OK ]
```

Para comprobar que la instalación ha sido correcta y se han realizado de forma satisfactoria la ejecución de los comandos, bastará comprobar la salida de la siguiente instrucción:

Código 6.36: Comprobación instalación del servidor ldap

```
mailuser@mail:~# sudo ldapsearch -LLL -W -D "cn=admin,dc=example,dc=com" \
-b "dc=example,dc=com" "(objectclass=*)"
Enter LDAP Password:
dn: dc=example,dc=com
objectClass: top
objectClass: dcObject
objectClass: organization
o: example.com
dc: example

dn: cn=admin,dc=example,dc=com
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
userPassword:: e1NTSEF9T3VNa0xjU0l3cmg5ZDcvbXdCVDFCaFN0ZXd6OE9PUlA=
```

La aplicación *phpldapadmin* permitirá una mejor administración del árbol de directorio LDAP, así como ofrecerá una interfaz más intuitiva para su manejo por personas menos diestras en las tareas de administración mediante consola. Su instalación se realizará con el comando *apt-get*, y una vez instalado bastará desde un explorador web acceder a la dirección *http://<ip-servidor>/phpldapadmin*.

Para terminar la configuración del árbol LDAP, será necesaria la creación de ciertos grupos, así como la inclusión de esquemas que nos permitan un mejor manejo del árbol. Para ello desde la propia herramienta de importación de objetos que ofrece *phpldapadmin* o desde consola con el comando *slapadd -v -l fichero.ldif* deberemos incluir los siguientes objetos en formato ldif.

Código 6.37: Ldif para usuarios

```
dn: ou=users,dc=example,dc=com
objectClass: top
objectClass: organizationalUnit
ou: users
description: Usuarios
```

Código 6.38: Ldif para grupos

```
dn: ou=groups,dc=example,dc=com
description: Grupos
objectclass: top
objectclass: organizationalUnit
ou: groups
```

Una vez creados dichos objetos, se podrá comenzar a incluir los grupos y usuarios necesarios para la correcta distribución del árbol LDAP. Es importante notar la distribución del árbol, ya que será necesario conocer de antemano la estructura para evitar errores a la hora de coordinar los distintos servicios que acceden al árbol. Con el comando *ldapsearch -LLL -W -D "cn=admin,dc=example,dc=com" -b "dc=example,dc=com" "(objectclass=*)"* puede volver a comprobarse que las anteriores inserciones han sido correctas, ya que de la salida del mismo se extrae toda la información relativa al directorio LDAP.

Además y para realizar las pruebas de conexión entre los distintos servicios, será necesario crear tanto un grupo de ejemplo como un usuario, pueden utilizarse los siguientes ldif de ejemplo. Primero habrá que crear el grupo y una vez creado el usuario correspondiente.

Código 6.39: Ldif para grupo users

```
dn: cn=users,ou=groups,dc=example,dc=com
cn: users
gidnumber: 500
objectclass: posixGroup
objectclass: top
```

Código 6.40: Ldif para usuario smontoiro

```
dn: cn=Sergio Montoiro,ou=users,dc=example,dc=com
cn: Sergio Montoiro
gidnumber: 500
givenname: Sergio
homedirectory: /home/users/smontoiro
loginshell: /bin/sh
mail: smontoiro@example.com
objectclass: inetOrgPerson
objectclass: posixAccount
objectclass: top
sn: Montoiro
telephonenumber: 668854922
uid: smontoiro
uidnumber: 1000
userpassword: {SSHA}3r6i5MJ6rxkO7ba3J4lkCBSF3ClUBeE2
```

Si en este momento se ejecutase el comando para comprobar la estructura del árbol, la salida sería parecida a esta:

Código 6.41: Comprobación instalación del servidor ldap

```
mailuser@mail:~# sudo ldapsearch -LLL -W -D "cn=admin,dc=example,dc=com" \
-b "dc=example,dc=com" "(objectclass=*)"
Enter LDAP Password:
dn: dc=example,dc=com
objectClass: top
objectClass: dcObject
objectClass: organization
o: example.com
dc: example

dn: cn=admin,dc=example,dc=com
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
userPassword:: e1NTSEF9T3VNa0xjU0l3cmg5ZDcvbXdCVDFCaFNOZXd6OE9PUlA=

dn: ou=users,dc=example,dc=com
objectClass: top
objectClass: organizationalUnit
ou: users
description: Usuarios

dn: ou=groups,dc=example,dc=com
objectClass: top
objectClass: organizationalUnit
ou: groups
description: Grupos

dn: cn=users,ou=groups,dc=example,dc=com
gidNumber: 500
cn: users
objectClass: posixGroup
objectClass: top
```

```
dn: cn=Sergio Montoiro,ou=users,dc=example,dc=com
cn: Sergio Montoiro
givenName: Sergio
gidNumber: 500
homeDirectory: /home/users/smontoiro
sn: Montoiro
loginShell: /bin/sh
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: top
userPassword:: e1NTSEF9M3I2aTVNSjZyeGtPN2JhM0o0bGtDQlNGM0NsVUJlRTI=
uidNumber: 1000
uid: smontoiro
mail: smontoiro@example.com
telephoneNumber: 668854922
```

Si el servidor fuese accedido desde el exterior, deberían incluirse las reglas necesarias en el cortafuegos para su acceso y utilización, véase Seguridad en página 125. En dicho caso sería recomendable utilizar el protocolo LDAP sobre SSL, para ello habría que crear un certificado específico para el servicio según las instrucciones dadas en el apartado Creación de certificados en la página 136, una vez creado dicho certificado, podría utilizarse en la configuración del servidor OpenLdap junto a la clave privada específica.

6.3.4. Servidor DNS

El servidor DNS elegido para su instalación será Djbdns. Para instalar este servidor de DNS bastará con seguir las siguientes indicaciones:

- Instalar los paquetes necesarios para su utilización:

Código 6.42: Instalación servidor dns

```
mailuser@mail:~# sudo apt-get install daemontools \
daemontools-run ucspi-tcp djbdns
```

- Una vez instalados los paquetes, se deberá crear los usuarios por defecto que utilizarán estos

servicios, para así añadir más seguridad al servidor. Los datos a introducir de estos usuarios son los de por defecto.

Código 6.43: Usuarios para el servidor dns

```
mailuser@mail:~# sudo adduser --no-create-home \
--disabled-login --shell /bin/false tinydns
mailuser@mail:~# sudo adduser --no-create-home \
--disabled-login --shell /bin/false dnslog
```

- Se configurará el servidor DNS para que escuche en la dirección por defecto del servidor, en este caso concreto *192.168.82.11*

Código 6.44: Configuración dirección IP de escucha

```
mailuser@mail:~# sudo tinydns-conf tinydns dnslog \
/etc/tinydns/ 192.168.82.11
```

- Por último, se configurará el servicio para que actúe como demonio, y esté disponible en el arranque del sistema operativo.

Código 6.45: Configuración del modo servicio

```
mailuser@mail:~# sudo mkdir -p /etc/service
mailuser@mail:~# sudo cd /etc/service
mailuser@mail:~# sudo ln -sf /etc/tinydns/
mailuser@mail:~# initctl start svscan
```

- Para comprobar que el servicio está correctamente levantado, bastará con ejecutar el siguiente comando.

Código 6.46: Comprobar si el servicio está ejecutando

```
mailuser@mail:~# sudo svstat /etc/service/tinydns
/etc/service/tinydns: up (pid 3602) 4 seconds
```

Para añadir entradas al servidor DNS, bastará con navegar a la carpeta */etc/tinydns/root* y ejecutar los script que ahí se encuentran. Se añadirán los servicios de correo, correo web, dns, ldap y cualquier otro servicio que se considere necesario.

Código 6.47: Añadir entradas a los ficheros de dns

```
mailuser@mail:~# cd /etc/tinydns/root
mailuser@mail:/etc/tinydns/root# sudo ./add-ns \
.example.com 192.168.82.11
mailuser@mail:/etc/tinydns/root# sudo ./add-host \
mail.example.com 192.168.82.11
mailuser@mail:/etc/tinydns/root# sudo ./add-alias \
webmail.example.com 192.168.82.11
mailuser@mail:/etc/tinydns/root# sudo ./add-alias \
pop.example.com 192.168.82.11
mailuser@mail:/etc/tinydns/root# sudo ./add-alias \
smtp.example.com 192.168.82.11
mailuser@mail:/etc/tinydns/root# sudo ./add-alias \
imap.example.com 192.168.82.11
mailuser@mail:/etc/tinydns/root# sudo ./add-alias \
ldap.example.com 192.168.82.11
```

La primera línea en la que se añade el el dominio con el comando *add-ns* es muy importante, ya que si no se especifica el dominio adecuado, el servidor DNS no devolverá los datos correctamente, e incluso fallará cualquier petición que se le haga.

Cualquier otro nombre de servicio que fuese necesario dar de alta, se deberían seguir las instrucciones dadas en las líneas anteriores. Una vez realizados los cambios, habría que construir el fichero DNS con el comando *make* y reiniciar el servicio, para ello bastará con ejecutar:

Código 6.48: Reconfiguración y reinicio del servidor dns

```
mailuser@mail:/etc/tinydns/root# sudo make; sudo svc -d \
/etc/service/tinydns; sleep 5;sudo svc -u \
/etc/service/tinydns
```

Para comprobar que el servidor está funcionando correctamente se puede utilizar el comando *dig* de la siguiente forma:

Código 6.49: Comprobar el servidor dns

```
mailuser@mail:~# dig @192.168.82.11 mail.example.com ANY

; <<>> DiG 9.7.3 <<>> @192.168.82.11 mail.example.com ANY
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 31140
;; flags: qr aa rd; QUERY: 1, ANSWER: 2, AUTHORITY: 1, ADDITIONAL: 2
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;mail.example.com.              IN      ANY

;; ANSWER SECTION:
mail.example.com.       86400   IN      MX      0 a.mx.mail.example.com.
mail.example.com.       86400   IN      A       192.168.82.11

;; AUTHORITY SECTION:
example.com.            259200  IN      NS      a.ns.example.com.

;; ADDITIONAL SECTION:
a.mx.mail.example.com.  86400   IN      A       192.168.82.11
a.ns.example.com.       259200  IN      A       192.168.82.11

;; Query time: 30 msec
;; SERVER: 192.168.82.11#53(192.168.82.11)
;; WHEN: Fri Apr  5 23:46:32 2013
;; MSG SIZE  rcvd: 122
```

Por último y para que el servicio esté correctamente funcionando y sea accesible desde la red, habrá que habilitar los puertos *UPD* correspondientes en el cortafuegos, en este caso sería necesario el 53. Véase Seguridad en página 125.

6.4. Servidor de correo

El siguiente servicio a instalar será el referente al servidor de correo. El servidor de correo está compuesto por diferentes programas que requieren de su propia configuración para que se lleve a cabo la compleja comunicación que debe existir entre todos ellos. El servidor de correo, se compone de distintas partes que deben hablarse entre si, y su configuración se debe realizar del siguiente modo.

6.4.1. Postfix y Dovecot

El servidor Postfix es el Mail Transport Agent (MTA) encargado de almacenar y reenviar los correos electrónicos entre los distintos usuarios mediante el protocolo Simple Mail Transfer Protocol (SMTP), mientras que Dovecot será el encargado de hacer accesible el sistema de forma remota, soportando los protocolos Internet Message Access Protocol (IMAP) y Post Office Protocol (POP). Su instalación se ejecuta igual que el resto de los servicios instalados en este proyecto, con el comando *apt-get*.

Código 6.50: Instalación de los paquetes asociados al servidor de correo

```
mailuser@mail:~# sudo apt-get install mail-server^
mailuser@mail:~# sudo apt-get install postfix-ldap dovecot-ldap
```

En el diálogo de configuración del servidor de correo seleccionar, sitio de Internet, el nombre del dominio (FQDN) en este ejemplo será *example.com*. La configuración que tendrá lugar en estos momento será la establecida por la distribución, a la cual se deberán añadir los ajustes necesarios para que se adecuen al entorno en el que se esté instalando el servidor.

La primera configuración que se llevará a cabo, será la de proponer un usuario virtual para el manejo de las carpetas del correo, que tenga los permisos necesarios para almacenar el correo en formato maildir, de esta forma la administración de los datos referentes al correo se realizará de forma más eficiente.

6.4.2. Configuración maildir

Los primeros pasos que se llevarán a cabo en la configuración del correo, corresponderán a la creación de un usuario que sea el encargado de administrar las carpetas donde estarán contenidos los buzones de correo de los usuarios finales. Para ello se creará un usuario denominado *vmail*, al que se le darán los permisos necesarios para realizar las operaciones de almacenamiento de los correos. Dicho usuario será utilizado cada vez que el servidor MTA decida crear un nuevo buzón, así como almacenar la información y los correos referentes al usuario.

Para crear el usuario y las posteriores configuraciones, se ejecutarán los siguientes comandos:

Código 6.51: Creación de los directorios virtuales

```
mailuser@mail:~# sudo useradd -r -u 150 -g mail -d /var/vmail \
-s /sbin/nologin -c "Virtual␣maildir␣handler" vmail
mailuser@mail:~# sudo mkdir /var/vmail
mailuser@mail:~# sudo chmod 777 /var/vmail
mailuser@mail:~# sudo chown vmail.mail /var/vmail
```

Con el primer comando se crea el usuario *vmail*, que pertenece al grupo *mail*, su directorio por defecto estará ubicado en */var/vmail*, y además, dicho usuario no podrá utilizar un entorno de consola, ya que no dispondrá de los permisos adecuados. Se creará su directorio de trabajo, y se le asignarán los permisos correspondientes.

De esta forma se asegura de que todos los usuarios tendrán su propio directorio de correo dentro de esta carpeta, y dichos directorios serán administrados por los servicios de correo, que utilizarán para ello el usuario *vmail*.

Para el correcto funcionamiento de las carpetas virtuales por parte del sistema operativo hay que configurar los métodos de autenticación local para que puedan utilizar el directorio LDAP. Para ello y después de instalar los siguientes paquetes:

Código 6.52: Autenticación local contra ldap

```
mailuser@mail:~# sudo apt-get install ldap-utils\
 libpam-ldap libnss-ldap nslcd
```

Donde se responderá con la información relativa al directorio local, incluyendo aquellos datos sobre la raíz del directorio, *dc=example,dc=com*, en su versión 3, y con el usuario y contraseña de

administración.

Además hay que modificar los ficheros relativos a los módulos de autenticación, del inglés Pluggable Authentication Modules (PAM), para ello bastará con editar el fichero:

Código 6.53: /etc/nsswitch.conf

```
...
passwd:         compat ldap
group:          compat ldap
shadow:         compat ldap
...
```

Añadiendo la opción de LDAP a las opciones de *passwd*,*group* y *shadow*.

Por último, para que el usuario vmail tenga los permisos necesarios para ejecutar los comandos referentes a Dovecot se deberá modificar el fichero */etc/sudoers*, con el comando *visudo* y añadir al final.

Código 6.54: /etc/sudoers

```
...
vmail    ALL=NOPASSWD:/usr/lib/dovecot/dovecot-lda
```

6.4.3. Configuración Postfix

Para la configuración de postfix, será necesario modificar varios ficheros contenidos bajo el directorio */etc/postfix*. En dicho directorio se concentran todos los archivos necesarios para el correcto funcionamiento del servicio de correo.

El primer archivo a editar será */etc/postfix/main.cf*, el fichero principal de configuración de postfix. La instalación del paquete realiza una pequeña modificación del fichero para funcionar tal como viene. Para configurar todas las opciones deseadas para el correcto funcionamiento, se expondrá el fichero en extractos del mismo, con un comentario al respecto del conjunto de líneas expuestas. Por problemas de espacio se indican saltos de línea en los ficheros con el carácter \, en la configuración de los ficheros habrá que eliminar dicho carácter y dicho salto de línea.

Código 6.55: /etc/postfix/main.cf (parte 1)

```
# The first text sent to a connecting process.
smtpd_banner = $myhostname ESMTP $mail_name
biff = no
append_dot_mydomain = no
readme_directory = no

# SASL parameters
# ---------------------------------

# Use Dovecot to authenticate.
smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth
smtpd_sasl_auth_enable = yes
broken_sasl_auth_clients = yes
smtpd_sasl_security_options = noanonymous
smtpd_sasl_local_domain =
smtpd_sasl_authenticated_header = yes

# TLS parameters
# ---------------------------------

smtpd_tls_cert_file=/etc/ssl/certs/mail.example.com.cert
smtpd_tls_key_file=/etc/ssl/private/mail.example.com.key
smtpd_tls_CAfile=/etc/ssl/local/cacert.pem
smtpd_use_tls=yes
smtp_tls_security_level = may
smtpd_tls_security_level = may
smtp_tls_note_starttls_offer = yes
smtpd_tls_loglevel = 1
smtpd_tls_received_header = yes
smtpd_tls_session_cache_timeout = 3600s
tls_random_source = dev:/dev/urandom
```

En este primer extracto se indica un pequeño comentario sobre el servidor, así como distintas

opciones por defecto de Postfix. En los parámetros que se refieren a la autenticación, Postfix delegará toda la autenticación al servidor Dovecot, no dejando entrar a usuarios sin autenticar.

Además se definen los parámetros para las conexiones sobre SSL. Se especifican los ficheros que contienen el certificado del servidor, su clave privada, así como el certificado de la Certification Authority (CA) que ha firmado dichos certificados. Se insta al servidor a utilizar conexiones seguras, y una serie de parámetros relativos a tiempos de conexión y que semilla utilizará para crear números aleatorios.

Código 6.56: /etc/postfix/main.cf (parte 2)

```
# will it be a permanent error or temporary
unknown_local_recipient_reject_code = 450
# how long to keep message on queue before return as failed.
# some have 3 days, I have 16 days as I am backup server for some people
# whom go on holiday with their server switched off.
maximal_queue_lifetime = 7d
# max and min time in seconds between retries if connection failed
minimal_backoff_time = 1000s
maximal_backoff_time = 8000s
# how long to wait when servers connect before receiving rest of data
smtp_helo_timeout = 60s
# how many address can be used in one message.
# effective stopper to mass spammers, accidental copy in whole address list
# but may restrict intentional mail shots.
smtpd_recipient_limit = 16
# how many error before back off.
smtpd_soft_error_limit = 3
# how many max errors before blocking it.
smtpd_hard_error_limit = 12
```

Estos parámetros de configuración se explican por si solos, y no se entrará al detalle. Cada servidor deberá llevar los que se establezcan en el momento de la instalación, y si fuese necesario se realizarían las modificaciones pertinentes.

Código 6.57: /etc/postfix/main.cf (parte 3)

```
# Requirements for the HELO statement
smtpd_helo_restrictions = permit_mynetworks, warn_if_reject \
reject_non_fqdn_hostname, reject_invalid_hostname, permit
# Requirements for the sender details
smtpd_sender_restrictions = permit_sasl_authenticated, \
permit_mynetworks, warn_if_reject reject_non_fqdn_sender, \
reject_unknown_sender_domain, reject_unauth_pipelining, permit
# Requirement for the recipient address.
smtpd_recipient_restrictions = reject_unauth_pipelining, \
permit_mynetworks, permit_sasl_authenticated, \
reject_non_fqdn_recipient, reject_unknown_recipient_domain, \
reject_unauth_destination, check_policy_service \
inet:127.0.0.1:10023, permit
# Requirements for the connecting server
smtpd_client_restrictions = reject_rbl_client sbl.spamhaus.org, \
reject_rbl_client blackholes.easynet.nl, reject_rbl_client \
dnsbl.njabl.org
smtpd_data_restrictions = reject_unauth_pipelining

# require proper helo at connections
smtpd_helo_required = yes
# waste spammers time before rejecting them
smtpd_delay_reject = yes
disable_vrfy_command = yes
```

Este tipo de opciones son muy importantes para la seguridad, algunas incluso dependen de otros servicios como los de Antivirus y Antispam que se detallarán en la siguiente sección. La configuración de estos parámetros advierte al servidor quiénes y desde que direcciones, tanto en nombre, como desde que ip, se pueden enviar correos a través de este MTA.

El primero de las opciones indica que si la comunicación se inicia desde un ordenador que no cumple unos mínimos de confianza, como puede ser el nombre del que inicia la conexión, y que este sea válido, el servidor rechazará la conexión.

En el segundo se insta a que el remitente esté autenticado o esté en la misma subred del servidor

para poder enviar correos. Además y como se ve en el tercer parámetro, si el receptor del correo no está correctamente formado, el correo será también rechazado.

El siguiente, comprobará si los servidores o clientes que inician la conexión vienen de direcciones indicadas en las distintas listas negras de internet, y en su caso serán rechazados.

Por último se especifican una serie de parámetros que ayudarán a mantener el correo libre de mensajes no deseados.

Código 6.58: /etc/postfix/main.cf (parte 4)

```
# General host and delivery info

myhostname = mail.example.com
myorigin = $myhostname
mydestination = $myhostname localhost
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
mynetworks_style = host

# Tell postfix to hand off mail to the definition for dovecot
virtual_transport = dovecot
dovecot_destination_recipient_limit = 1

# Getting rid of unwanted headers.
header_checks = regexp:/etc/postfix/header_checks
# getting rid of x-original-to
enable_original_recipient = no
```

Diferentes configuraciones para el conocimiento completo de la situación de la red por parte de Postfix, de esta forma el servidor conocerá como enviar los correos, y como enrutar aquellos que no administre.

En las siguientes líneas se especifica la conexión con Dovecot, así como la limitación para que sólo se puede enviar a un usuario por buzón. Un par de filtros para las cabeceras, finalizan esta parte de la configuración.

Código 6.59: /etc/postfix/main.cf (parte 5)

```
# Use amavis for virus and spam scanning
content_filter = amavis:[127.0.0.1]:10024

############## LDAP

virtual_mailbox_base= /var/vmail
virtual_mailbox_maps = ldap:/etc/postfix/ldap/mailboxes.cf
virtual_alias_maps = ldap:/etc/postfix/ldap/virtual_groups.cf, \
ldap:/etc/postfix/ldap/virtual_aliases.cf
virtual_mailbox_domains = ldap:/etc/postfix/ldap/virtual_domains.cf
virtual_mailbox_base = /home/domains
virtual_uid_maps = ldap:/etc/postfix/ldap/users_uid.cf
```

Esta es la última parte del fichero, en él se especifica la conexión con el servicio de antivirus. Y por último la conexión contra el servidor LDAP. En dicha configuración sólo se especifican los ficheros que tendrá que leer para continuar con la configuración de Postfix por lo que se seguirán detallando en este apartado. Se puede decir, que en cada uno de los ficheros irá la información referente a cómo autenticar los usuarios, cómo extraer su nombre, la información referente al mismo, si existen grupos de usuarios (útil para alias de correo), etc.

El siguiente fichero a editar será */etc/postfix/master.cf*. Dicho fichero contiene todos los servicios con los que puede funcionar Postfix. El fichero por defecto aparecerá totalmente comentado, en las siguientes líneas tan sólo se especificará un extracto del mismo, ya que su contenido es bastante extenso.

Lo primero que hay que hacer es descomentar la línea que comienza por smtps y especificar las siguientes como se indica a continuación:

De esta forma arrancamos el servicio de SMTP sobre SSL e incluimos las opciones para prevenir fallos de seguridad y ataques no deseados. Además se incluirán bajo la línea pickup distintas opciones para aumentar la seguridad.

Código 6.60: /etc/postfix/master.cf (parte 1)

```
smtps     inet  n       -       -       -       -       smtpd
  -o syslog_name=postfix/smtps
  -o smtpd_tls_wrappermode=yes
  -o smtpd_sasl_auth_enable=yes
  -o smtpd_tls_auth_only=yes
  -o smtpd_client_restrictions=permit_sasl_authenticated,\
  reject_unauth_destination,reject
  -o smtpd_sasl_security_options=noanonymous,noplaintext
  -o smtpd_sasl_tls_security_options=noanonymous
#628       inet  n       -       -       -       -       qmqpd
pickup    fifo  n       -       -       60      1       pickup
  -o content_filter=
  -o receive_override_options=no_header_body_checks
```

Código 6.61: /etc/postfix/master.cf (parte 2)

```
#
# The next two entries integrate with Amavis for anti-virus/spam checks.
#
amavis      unix    -       -       -       -       2       smtp
        -o smtp_data_done_timeout=1200
        -o smtp_send_xforward_command=yes
        -o disable_dns_lookups=yes
        -o max_use=20
127.0.0.1:10025 inet    n       -       -       -       -       smtpd
        -o content_filter=
        -o local_recipient_maps=
        -o relay_recipient_maps=
        -o smtpd_restriction_classes=
        -o smtpd_delay_reject=no
        -o smtpd_client_restrictions=permit_mynetworks,reject
        -o smtpd_helo_restrictions=
        -o smtpd_sender_restrictions=
        -o smtpd_recipient_restrictions=permit_mynetworks,reject
        -o smtpd_data_restrictions=reject_unauth_pipelining
        -o smtpd_end_of_data_restrictions=
        -o mynetworks=127.0.0.0/8
        -o smtpd_error_sleep_time=0
        -o smtpd_soft_error_limit=1001
        -o smtpd_hard_error_limit=1000
        -o smtpd_client_connection_count_limit=0
        -o smtpd_client_connection_rate_limit=0
        -o receive_override_options=no_header_body_checks,\
        no_unknown_recipient_checks



#
# Integration with Dovecot - hand mail over to it for local delivery, and
# run the process under the vmail user and mail group.
#
dovecot     unix    -       n       n       -       -   pipe
        flags=DRhu user=vmail:mail argv=/usr/bin/sudo\
         /usr/lib/dovecot/dovecot-lda -f \${sender} -d \${user}
```

La conexión con otros servicios se especificará con las líneas anteriores, que se podrán añadir al final del fichero. En dichas líneas se especifica la conexión con el software antivirus, así como con Dovecot. Para este último se utiliza el comando sudo, es por ello que en el código visto en la página 154 se añadió la entrada correspondiente para la no utilización de contraseñas para el usuario *vmail* con el comando indicado.

Código 6.62: /etc/postfix/dynamicmaps.cf

```
# Postfix dynamic maps configuration file.
#
#type    location of .so file                    open function    (mkmap func)
#====    ================================        =============    ============
tcp              /usr/lib/postfix/dict_tcp.so            dict_tcp_open
sqlite   /usr/lib/postfix/dict_sqlite.so         dict_sqlite_open
ldap     /usr/lib/postfix/dict_ldap.so       dict_ldap_open
```

Para poder conectarse contra LDAP, será necesario cargar el módulo correspondiente. Para ello bastará con añadir la última línea al fichero */etc/postfix/dynamicmaps.cf*.

Además el fichero */etc/postfix/header_checks* deberá tener el siguiente contenido.

Código 6.63: /etc/postfix/header_checks

```
/^Received:/                    IGNORE
/^User-Agent:/                  IGNORE
/^X-Mailer:/                    IGNORE
/^X-Originating-IP:/            IGNORE
/^x-cr-[a-z]*:/                 IGNORE
/^Thread-Index:/                IGNORE
```

Los siguientes ficheros especifican la conexión entre el servidor de correo y el servidor LDAP. En cada uno de los ficheros está especificada la cadena de conexión contra el servidor, así como la solicitud de información. Para el árbol creado en la sección Servidor LDAP se especifican cada una de las solicitudes dependiendo del resultado que se espera obtener. Si el árbol creado es distinto al descrito en este documento, los parámetros podrían variar.

De los valores resultantes de cada una de las solicitudes, se especifica al servidor con que parámetro propio lo tiene que mapear. De ese modo se consigue extraer la información necesaria del servidor.

Código 6.64: /etc/postfix/ldap/mailboxes.cf

```
server_host = ldap://localhost:389
bind_dn = cn=admin,dc=example,dc=com
bind_pw = m41l9s3r
search_base = dc=com
version = 3
timeout = 5
debug = 256
scope = sub
bind = yes
start_tls = no
tls_require_cert = no
query_filter = (&(objectClass=posixAccount)\
(accountStatus=active)(|(mail=%s)(mailAlternateAddress=%s)))
result_attribute = mail
result_format = %d/%u/
```

Código 6.65: /etc/postfix/ldap/virtual_groups.cf

```
server_host = ldap://localhost:389
bind_dn = cn=admin,dc=example,dc=com
bind_pw = m41l9s3r
search_base = dc=example,dc=com
version = 3
timeout = 5
debug = 256
scope = sub
bind = yes
start_tls = no
tls_require_cert = no
query_filter = (&(objectClass=posixAccount)\
(|(mail=%s)(mailAlternateAddress=%s)))
result_attribute = mail
result_format = %u@%d
```

Código 6.66: /etc/postfix/ldap/virtual_aliases.cf

```
server_host = ldap://localhost:389
bind_dn = cn=admin,dc=example,dc=com
bind_pw = m41l9s3r
search_base = dc=example,dc=com
version = 3
timeout = 5
debug = 256
scope = sub
bind = yes
start_tls = no
tls_require_cert = no
query_filter = (&(objectClass=inetLocalMailRecipient)\
(|(mail=%s)(mailAlternateAddress=%s)))
result_attribute = mailLocalAddress
result_format = %u@%d
```

Código 6.67: /etc/postfix/ldap/virtual_domains.cf

```
server_host = ldap://localhost:389
bind_dn = cn=admin,dc=example,dc=com
bind_pw = m41l9s3r
search_base = dc=example,dc=com
version = 3
timeout = 5
debug = 256
scope = sub
bind = yes
start_tls = no
tls_require_cert = no
query_filter = (&(objectClass=dcObject)(objectClass=organization)(o=%s))
result_attribute = o
```

Código 6.68: /etc/postfix/ldap/users_uid.cf

```
server_host = ldap://localhost:389
bind_dn = cn=admin,dc=example,dc=com
bind_pw = m41l9s3r
search_base = dc=com
version = 3
timeout = 5
debug = 256
scope = sub
bind = yes
start_tls = no
tls_require_cert = no
query_filter = (&(objectClass=posixAccount)\
(accountStatus=active)(|(mail=%s)(mailAlternateAddress=%s)))
result_attribute = uidNumber
```

6.4.4. Configuración Dovecot

Al igual que Postfix y la gran mayoría de servicios de linux, Dovecot tiene su configuración bajo el directorio */etc*, exactamente en */etc/dovecot*. El primer fichero que se deberá modificar, es el fichero general de configuración, */etc/dovecot/dovecot.conf*. La configuración que viene especificada hace que el servidor tan sólo esté activado para la dirección local y con valores por defecto que no ofrecerán todas la características que se requieren de este servicio. Las líneas que se descomentarán en este fichero para su correcto funcionamiento serán las siguientes.

Código 6.69: /etc/dovecot/dovecot.conf

```
...
listen = 192.168.82.11
...
base_dir = /var/run/dovecot/
...
```

De esta forma se configura el servicio para que escuche en la interfaz de red que estará expuesta al exterior, y además se especifica el directorio de trabajo del servicio.

Para que el servicio Dovecot funcione correctamente contra el servidor LDAP se requiere configurar una serie de ficheros que se encuentran dentro de la carpeta */etc/dovecot/conf.d*. Dichos ficheros se resumen a continuación. La mayoría de los ficheros vienen con todas las opciones comentadas, en estos ejemplos se especificará con puntos suspensivos (...) aquellas líneas que se dejan exactamente igual que la configuración por defecto de la distribución del SO.

Código 6.70: /etc/dovecot/conf.d/10-auth.conf

```
...
disable_plaintext_auth = yes
...
auth_mechanisms = plain login
...
#!include auth-system.conf.ext
#!include auth-sql.conf.ext
!include auth-ldap.conf.ext
...
```

En el fichero de autenticación, se ha descomentado la opción para deshabilitar la autenticación en claro, por lo que sólo se podrán utilizar contraseñas codificadas con algún mecanismo de resumen. En la última parte se especifica que se incluya la configuración del fichero *auth-ldap.conf.ext* que contendrá la información de la autenticación contra LDAP.

Código 6.71: /etc/dovecot/conf.d/10-master.conf

```
...
service auth {
  # auth_socket_path points to this userdb socket by default. It's typically
  # used by dovecot-lda, doveadm, possibly imap process, etc. Its default
  # permissions make it readable only by root, but you may need to relax these
  # permissions. Users that have access to this socket are able to get a list
  # of all usernames and get results of everyone's userdb lookups.
  unix_listener auth-userdb {
    mode = 0600
    user = vmail
    group = mail
  }

  # Postfix smtp-auth
  unix_listener /var/spool/postfix/private/auth {
    mode = 0666
        user = postfix
        group = postfix
  }
...
```

En el fichero *10-master.conf* se le indica al servidor que servicios va a tener disponibles, en que puertos y con que opciones se van a levantar dichos servicios. Las líneas modificadas, se refieren a los permisos de Dovecot con el sistema de ficheros, además de su conexión con el servicio Postfix.

Código 6.72: /etc/dovecot/conf.d/auth-ldap.conf.ext

```
# Authentication for LDAP users. Included from auth.conf.
#
# <doc/wiki/AuthDatabase.LDAP.txt>

passdb {
  driver = ldap

  # Path for LDAP configuration file, see example-config/dovecot-ldap.conf.ext
  args = /etc/dovecot/conf.d/dovecot-ldap.conf.ext
}

userdb {
  driver = ldap
  args = /etc/dovecot/conf.d/dovecot-ldap.conf.ext
}
```

Este fichero será creado para la configuración específica que deberá utilizar Dovecot contra el servidorLDAP. En el se indica en que fichero se encuentran las cadenas de conexión, el usuario y contraseña para obtener la información, etc.

Código 6.73: /etc/dovecot/conf.d/dovecot-ldap.conf.ext(parte 1)

```
# Space separated list of LDAP hosts to use. host:port is allowed too.
hosts = localhost
# Distinguished Name - the username used to login to the LDAP server.
# Leave it commented out to bind anonymously (useful with auth_bind=yes).
dn = uid=admin,dc=example,dc=com
# Password for LDAP server, if dn is specified.
dnpass = m41l9s3r
# LDAP protocol version to use. Likely 2 or 3.
ldap_version = 3
```

Código 6.74: /etc/dovecot/conf.d/dovecot-ldap.conf.ext(parte 2)

```
# LDAP base. %variables can be used here.
# For example: dc=mail, dc=example, dc=org
base = ou=users, dc=example,dc=com

# Search scope: base, onelevel, subtree
scope = subtree
# There are also other special fields which can be returned, see
# http://wiki2.dovecot.org/UserDatabase/ExtraFields
user_attrs = homeDirectory=home,uidNumber=uid,gidNumber=gid, mail_plugins

# Filter for user lookup. Some variables can be used (see
# http://wiki2.dovecot.org/Variables for full list):
#   %u - username
#   %n - user part in user@domain, same as %u if there's no domain
#   %d - domain part in user@domain, empty if user there's no domain
user_filter = (&(objectClass=posixAccount)(uid=%u))

# Password checking attributes:
#  user: Virtual user name (user@domain), if you wish to change the
#        user-given username to something else
#  password: Password, may optionally start with {type}, eg. {crypt}
# There are also other special fields which can be returned, see
# http://wiki2.dovecot.org/PasswordDatabase/ExtraFields
pass_attrs = uid=user,userPassword=password
# Filter for password lookups
pass_filter = (&(objectClass=posixAccount)(uid=%u))
# Default password scheme.
# List of supported schemes is in: http://wiki2.dovecot.org/Authentication
default_pass_scheme = CRYPT
```

Para este fichero pueden llegar a utilizarse un gran número de parámetros referentes a la conexión con el servidor LDAP. Hay más información al respecto en esta dirección de internet http://wiki2.dovecot.org/AuthDatabase/LDAP. En resumen se indica que la conexión se va

a realizar contra el servidor local, utilizando las credenciales del administrador. Se va a obtener información de los usuarios que se encuentran bajo la rama *ou=users, dc=example,dc=com* y se indican las consultas de autenticación. En caso de tener un árbol LDAP configurado de forma distinta, se tiene que tener en cuenta dicha estructura al editar este fichero.

Código 6.75: /etc/dovecot/conf.d/10-mail.conf

```
...
mail_location = maildir:/var/vmail/%n
...
mail_uid = vmail
mail_gid = mail
...
```

El resto de parámetros se puede mantener comentado. Las variables modificadas se refieren al lugar donde van a estar contenidos los directorios de los usuarios, y el usuario y grupo del encargado de mantener dichas carpetas.

Código 6.76: /etc/dovecot/conf.d/15-lda.conf

```
...
protocol lda {
  # Space separated list of plugins to load (default is global mail_plugins).
  postmaster_address = postmaster@example.com
}
...
```

En este fichero se indica para el protocolo *lda* la dirección por defecto del usuario *postmaster*.

Código 6.77: /etc/dovecot/conf.d/10-ssl.conf

```
...
ssl_cert = </etc/ssl/certs/mail.example.com.cert
ssl_key = </etc/ssl/private/mail.example.com.key
...
```

Por último, en el fichero de configuración SSL se especifican los archivos que contienen tanto la clave privada como el certificado público para las conexiones ssl.

Antes de reiniciar todos los servicios y comprobar que todas las configuraciones se han realizado de forma correcta habrá que instalar otra parte muy importante del servicio de correo, el software Antivirus y Antispam.

Por último y para que el servicio esté correctamente funcionando y sea accesible desde la red, habrá que habilitar los puertos *TCP* correspondientes en el cortafuegos, en este caso serán necesarios los puertos 25, 110, 143, 465, 585, 993 y 995 para todos los servicios configurados. Véase Seguridad en página 125.

6.4.5. Antivirus y Antispam

Las instalación del servicio de Antivirus y Antispam, se realizará con los parámetros por defecto de la distribución. La configuración inicial de este paquete está suficientemente parametrizada para asegurar una protección eficaz de los correos electrónicos. En caso de necesitar una configuración extra, se recomienda la lectura de los manuales de los ficheros de configuración de los paquetes instalados, así como una vista a las páginas web correspondientes, en la Antivirus y Antispam hay más información al respecto.

Para instalar el software, bastará con ejecutar los siguientes comandos:

Código 6.78: Instalación de las herramientas antivirus y antispam

```
mailuser@mail:~# sudo apt-get install amavis clamav \
clamav-daemon spamassassin
mailuser@mail:~# sudo apt-get install pyzor razor arj
mailuser@mail:~# sudo apt-get install cabextract nomarch\
 unzip zip p7zip unrar-free lzop rpm2cpio zoo ripole
```

Se instalarán una serie de paquetes necesarios para el completo funcionamiento de los programas, la mayoría de ellos son programas utilizados para descomprimir y escanear archivos que se encuentran en los ficheros adjuntos.

Para una correcta sincronización entre el software antivirus y antispam, se añadirán los usuarios encargados de ejecutar los procesos al grupo del servicio contrario, es decir:

Código 6.79: Intercambio de grupos para clamav y amavis

```
mailuser@mail:~# sudo adduser clamav amavis
mailuser@mail:~# sudo adduser amavis clamav
```

Por último y para activar el antivirus Amavis hay que editar el siguiente fichero descomentando las líneas:

Código 6.80: /etc/amavis/conf.d/15-content_filter_mode

```
...
@bypass_virus_checks_maps = (
   \%bypass_virus_checks, \@bypass_virus_checks_acl,
   \$bypass_virus_checks_re);
...
```

Para utilizar correctamente el servicio de SpamAssassin hay que editar el fichero */etc/default/spamassasin* para que tenga un aspecto parecido a este:

Código 6.81: /etc/default/spamassasin

```
...
ENABLED=1
...
CRON=1
```

6.5. Servidor Groupware

El servidor groupware será suministrado a través del servidor web, y por lo tanto antes de comenzar con la instalación del mismo, se pueden llevar a cabo algunos retoques en los ficheros de configuración de PHP y Apache Web Server para mejorar la visibilidad y seguridad de los mismos.

Código 6.82: /etc/php5/apache2/php.ini

```
...
expose_php = Off
...
```

Código 6.83: /etc/apache2/conf.d/security

```
...
ServerTokens Prod
...
ServerSignature Off
```

De esta forma se consigue que la información que se muestra del servidor en las cabeceras HTTP, o en los mensajes de error, sean las mínimas para poder evitar ataques en caso de algún fallo de seguridad de la versión concreta, y que el atacante pueda explotarla.

Además se habilitarán las opciones para utilizar SSL por parte del servidor web, para ello bastará con ejecutar los comandos:

Código 6.84: Creación de los directorios virtuales

```
mailuser@mail:~# sudo a2enmod rewrite ssl
mailuser@mail:~# sudo a2ensite default-ssl
mailuser@mail:~# sudo service apache2 restart
```

De esta forma se habilitan las opciones SSL para Apache Web Server. Antes de realizar cualquier otra instalación, se modificarán los ficheros referentes a la configuración de los sitios por defecto del servidor.

Código 6.85: /etc/apache2/sites-available/default

```
RewriteEngine On
RewriteCond %{HTTPS} off
RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI}

<VirtualHost *:80>
        ServerAdmin webmaster@localhost

        DocumentRoot /var/www
        <Directory />
                Options FollowSymLinks
                AllowOverride None
        </Directory>

        ErrorLog ${APACHE_LOG_DIR}/error.log

        # Possible values include: debug, info, notice, warn, error, crit,
        # alert, emerg.
        LogLevel warn

        CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Código 6.86: /etc/apache2/sites-available/default-ssl

```
<VirtualHost mail.example.com:443>
                ...

        SSLEngine on
        SSSLCertificateFile     /etc/ssl/certs/mail.example.com.cert
        SSLCertificateKeyFile   /etc/ssl/private/mail.example.com.key

        ...
</VirtualHost>
```

En este último fichero tan sólo habrá que modificar los datos respecto a los archivos que contienen los certificados.

De esta forma, se configura el servidor para que toda las conexiones no seguras se redirijan hacia una segura, bajo SSL y utilizando los certificados creados en la sección Creación de certificados.

6.5.1. Horde IMP

Una vez preparado el servidor se instalarán la siguiente serie de paquetes para el correcto funcionamiento de Horde IMP.

Código 6.87: Prerrequisitos de Horde IMP

```
mailuser@mail:~# sudo apt-get install php5-dev php5-sasl\
 php-pear php5-tidy php5-imagick php5-geoip libgeoip1\
 geoip-bin geoip-database php-xml-serializer php5-memcache\
 php-soap php5-intl libidn11-dev libmagickwand-dev\
 libmagick++4 imagemagick libsasl2-dev libssh2-php\
 libphp-jpgraph php-http-webdav-server libimage-exiftool-perl
```

Entre estos paquetes se incluyen aquellos necesarios para el tratamiento de imágenes, creación y tratamiento de ficheros xml, generación de vistas previas, etc.

Una vez realizada la instalación de los prerrequisitos, se va a realizar la instalación del paquete Horde IMP, gracias a los comandos *pecl* y *pear*, los cuales significan, PHP Extension Community Library (pecl) y PHP Extension and Application Repository (pear). Con el siguiente comando actualizamos y añadimos los repositorios correspondientes:

Código 6.88: Canales para instalación de Horde IMP

```
mailuser@mail:~# sudo pecl channel-update pear.php.net
mailuser@mail:~# sudo pear channel-update pear.php.net
mailuser@mail:~# sudo pear channel-discover pear.horde.org
```

Y con los siguientes se procede a instalar los paquetes adecuados. Las versiones que se indician son las versiones más actuales de los paquetes, que puede que no coincidan con las versiones disponibles a la hora de realizar una instalación. Para ello habrá que buscar la información correspondiente en la página concreta de cada uno de los módulos.

Código 6.89: Prerrequisitos e instalación de Horde IMP

```
mailuser@mail:~# sudo pecl install lzf
mailuser@mail:~# sudo pear install --alldeps\
 channel://pear.php.net/Date_Holidays-0.21.8
mailuser@mail:~# sudo pear install --alldeps\
 channel://pear.php.net/Date_Holidays_UNO
mailuser@mail:~# sudo pear install --alldeps\
 channel://pear.php.net/Date_Holidays_Spain-0.1.3
mailuser@mail:~# sudo pear install --alldeps\
 channel://pear.php.net/Numbers_Words-0.16.4
mailuser@mail:~# sudo pear install --alldeps\
 channel://pear.php.net/Text_CAPTCHA-0.4.6
mailuser@mail:~# sudo pear install pear/Net_LDAP2
mailuser@mail:~# sudo pear install horde/Horde_role
```

Con esta serie de comandos, se instalarán las librerías y aplicaciones necesarias para el correcto funcionamiento del servidor groupware. Para continuar con la instalación se procederá a crear el directorio donde va a residir la aplicación, en este caso */var/www/webmail*, que luego será solicitado por el script de inicio de instalación. Dicho programa se invocará con los siguientes comandos.

Código 6.90: Instalación de Horde IMP

```
mailuser@mail:~# sudo pear run-scripts horde/Horde_Role
mailuser@mail:~# sudo pear install -a -B horde/webmail
```

El último comando tardará en ejecutarse un tiempo, ya que descarga e instala todas las dependencias y programas relativos a las herramientas de Horde. Para comenzar a utilizar desde este momento el servicio, bastará con realizar los siguientes pasos para cambiar los permisos al directorio donde está el servidor de correo web, incluir en el directorio de PHP la configuración para la utilización de *lzf*, y por último reiniciar el servidor web.

Código 6.91: Últimos pasos de configuración de Horde IMP

```
mailuser@mail:~# sudo chown -R www-data:www-data /var/www
mailuser@mail:~# sudo echo extension=lzf.so > /etc/php5/conf.d/lzf.ini
mailuser@mail:~# sudo service apache2 restart
```

En este momento ya se podrá acceder a la página web del servidor groupware, que en primera instancia nos redirigirá a la página de inicio de Horde, con el usuario administrador. Los primero que habrá que configurar es toda la parte de administración de las distintas herramientas que compone el servidor groupware, dicha configuración se puede realizar por dos vías, vía web, o por el contrario editando los ficheros de configuración de cada una de ellas.

Horde no recomienda la modificación de los ficheros de configuración, por lo tanto dicha configuración se realizará desde la interfaz web. Para entrar en la configuración bastará con acceder a través del icono de configuración de administración, o directamente en la url `https://mail.example.com/webmail/admin/config/`.

La mayoría de las herramientas estarán en este momento desactivadas, para poder activarlas, habrá que crear en primera instancia la configuración para la conexión con base de datos, para ello en la pestaña *Database* para la configuración de Horde, con la configuración que aparece por defecto, y si la base de datos está accesible a través del socket UNIX, no será necesaria una configuración adicional. En caso de ser necesario se creará la base de datos, y se darán los permisos de administración total sobre dicha base de datos al usuario configurado.

Código 6.92: Valores de configuración de Base de datos.

```
conf['sql']['phptype'] = 'mysql/PDO';
conf['sql']['persistent'] = true;
conf['sql']['username'] = 'horde';
conf['sql']['password'] = 'hordepassword';
conf['sql']['protocol'] = 'unix';
conf['sql']['socket'] = '/var/run/mysqld/mysqld.sock';
conf['sql']['database'] = 'horde';
conf['sql']['charset'] = 'utf-8';
conf['sql']['ssl'] = false;
conf['sql']['splitread'] = false;
```

Una vez creada la base de datos, se podrá continuar con cada una de las pestañas de configuración que en este caso deben ser modificadas, y que no aceptan los valores por defecto. Se especificará cada uno de los campos, así como los valores a consignar en cada uno de ellos.

Para LDAP los valores que se especificarán serán los siguientes que se ven a continuación. Para realizar la conexión se utilizará el usuario administrador, que recuperará todos los valores del árbol para rellenar la información pertinente en los perfiles del usuario.

Código 6.93: Valores de configuración de LDAP.

```
conf['ldap']['hostspec'] = 'localhost';
conf['ldap']['tls'] = false;
conf['ldap']['version'] = 3;
conf['ldap']['binddn'] = 'cn=admin,dc=example,dc=com';
conf['ldap']['bindpw'] = 'm41l9s3r';
conf['ldap']['bindas'] = 'admin';
conf['ldap']['useldap'] = true;
```

En la pestaña de autenticación, habrá que prestar especial atención a la configuración de los usuarios que serán administradores. Bastará con poner en el campo correspondiente el nombre de los usuarios separados por comas, de aquellos que tendrán permisos de administración. Además se configurarán todos los valores necesarios para que el servidor groupware pueda recuperar del árbol LDAP los objetos exactos para poder utilizarlos como usuario, en este caso, se hace necesaria la configuración de la autenticación LDAP, así como los parámetros que se verán a continuación:

Código 6.94: Valores de configuración de autenticación.

```
conf['auth']['admins'] = array('Administrator', 'smontoiro');
conf['auth']['checkip'] = true;
conf['auth']['checkbrowser'] = true;
conf['auth']['resetpassword'] = true;
conf['auth']['alternate_login'] = false;
conf['auth']['redirect_on_logout'] = false;
conf['auth']['list_users'] = 'list';
conf['auth']['params']['basedn'] = 'dc=example,dc=com';
conf['auth']['params']['scope'] = 'sub';
conf['auth']['params']['ad'] = false;
conf['auth']['params']['uid'] = 'uid';
conf['auth']['params']['encryption'] = 'crypt';
conf['auth']['params']['newuser_objectclass'] = array('posixAccount',
'inetOrgPerson');
conf['auth']['params']['filter'] = '(objectclass=posixAccount)';
conf['auth']['params']['password_expiration'] = 'no';
conf['auth']['params']['driverconfig'] = 'horde';
conf['auth']['driver'] = 'ldap';
conf['auth']['params']['count_bad_logins'] = true;
conf['auth']['params']['login_block'] = false;
conf['auth']['params']['login_block_count'] = 5;
conf['auth']['params']['login_block_time'] = 5;
```

El resto de parámetros de configuración se mantendrán en sus valores por defecto. Algunos de los módulos de Horde necesita de un espacio de almacenamiento para los datos de los usuarios, por defecto se mantendrá que el motor de base de datos sea el que almacene dicha información. Por ello en la configuración de las pestañas de Alarma o Grupos, se mantendrá como driver la base de datos sql. Una vez rellenados y revisados todos los campos se pulsará el botón *Generar configuración de Horde* y si todo está bien, en la parte inferior de la pantalla aparecerá una ventana emergente que indicará que los cambios han sido almacenados con éxito y que se puede continuar con la configuración.

De vuelta en la pantalla de configuración principal, y para terminar de configurar el resto de herramientas, el primer paso a realizar es la creación de los esquemas de las bases de datos para todas ellas. Dicho paso se puede realizar herramienta a herramienta, pero también es posible llevarlo a cabo desde el botón *Actualizar todos los esquemas de base de datos*, esto creará los esquemas necesarios para cada una de las aplicaciones.

La primera aplicación a configurar será la de correo web, o *imp*. Por defecto el usuario podrá crear desde la pantalla de su navegador web carpetas, y se le permitirá ver el código fuente de los mensajes. El resto de parámetros será dejado el de por defecto. Se mantendrá un log de los mensajes enviados en base de datos para futuras consultas.

Para conseguir que el usuario entre en su cuenta de correo a través de la aplicación en línea habrá que crear un archivo en la carpeta */var/www/webmail/imp/config* que será el siguiente:

Código 6.95: /var/www/webmail/imp/config/backends.local.php

```
<?php
$servers['imap']['hordeauth'] = true;
```

De esta forma se comunica al servidor Horde que utilice el mismo usuario y contraseña para autenticar al usuario contra el servidor de correo. En este caso y al estar almacenadas todas las contraseñas en el LDAP no existe ningún problema, y la autenticación se realiza con éxito.

El resto de herramienta y su configuración se mantiene con los valores por defecto. Si fuese necesario modificar cualquier valor se debería realizar desde la consola de administración. Es recomendable guardar los ficheros de configuración para futuras referencias así como copias de seguridad frecuentes de los mismos.

Antes de comprobar el correcto funcionamiento del sistema *Horde* se deberá modificar el fichero de configuración del servicio HTTP para que se puedan realizar las tareas de sincronización entre los dispositivos. Dichas tareas de sincronización se realizan al redirigir las peticiones usuales de los dispositivos contra servidores exchange. Estas peticiones son procesadas por uno de los servicios de *Horde* que realiza las oportunas tareas, y responde a los dispositivos de la forma correcta.

Código 6.96: /etc/apache2/sites-available/default-ssl

```
...

 </Directory>



Alias /Microsoft-Server-ActiveSync /var/www/webmail/rpc.php
Alias /autodiscover/autodiscover.xml /var/www/webmail/rpc.php

RewriteEngine On
RewriteRule .* - [E=HTTP_MS_ASPROTOCOLVERSION:%{HTTP:Ms-Asprotocolversion}]
RewriteRule .* - [E=HTTP_X_MS_POLICYKEY:%{HTTP:X-Ms-Policykey}]
RewriteRule .* - [E=HTTP_AUTHORIZATION:%{HTTP:Authorization}]

ErrorLog ${APACHE_LOG_DIR}/error.log
...
```

Una vez que todo ha sido configurado, y para comprobar el correcto funcionamiento de la herramienta puede accederse a la página test.php para comprobar que la instalación del servidor ha sido correcta. Pueden ejecutarse distintos test para las distintas herramientas como son imp, ingo, etc. Es recomendable ejecutar todos los test, así como ver los posibles errores y seguir las indicaciones del software para corregir los mismos.

6.6. Post instalación

Después de instalar todos los elementos, queda preparar el entorno para recuperarse ante imprevistos o catástrofes, ya sea por fallos de seguridad que comprometan los datos, fallos hardware que dejen inservibles los servicios, etc.

Para prevenir errores mayores, debe implementarse una buena política de copias de seguridad sobre todos los sistemas afectados. Dicha política de copias incluye tanto la realización de las copias como la restauración de los servicios a un estado estable. Para ello se definen los siguientes pasos a realizar en cuanto a copias de seguridad:

- Ficheros de configuración: una buena política de control de versiones para los ficheros de configuración [12], ayuda a mantener el sistema saneado, así como prevenir errores humanos. Realizar la conexión con un repositorio externo para mantener las versiones sería deseable. Los ficheros de configuración de este sistema se encuentran en los directorios */etc*, y */var/www/**/conf/*.conf*. Si es necesario añadir más ficheros, proceder siempre del mismo modo.

- Directorios de correo: los directorios de correo de los usuarios, tan sólo contienen los ficheros con cada uno de los correos de dicho usuario. Dependiendo del número de ficheros, una copia semanal total de todos los ficheros, intercalando copias incrementales diarias, debería ser suficiente para mantener entre 20 y 50 buzones de correo.

- Base de datos: Copias de seguridad de la base de datos deben realizarse semanalmente, al igual que lo comentado anteriormente. También, es recomendable intercalar copias incrementales diarias. Dependiendo de la criticidad del sistema, así como del uso del mismo, se pueden modificar los períodos de tiempo que transcurre entre las copias completas e incrementales, adecuando éstas a los requisitos de los servicios.

Con las tres piezas anteriores almacenadas en sistemas de copias de seguridad, se puede replicar el servidor siguiendo los pasos de instalación del mismo y restaurando las copias del siguiente modo:

- Ficheros de configuración: restaurar los ficheros desde la última versión del control de versiones, prestando especial atención aquellos en los que exista configuración explícita que sólo puede estar en una máquina, como puede dirección o nombre de la máquina, dirección IP de conexión con base de datos, etc.

- Directorios de correo: restaurar los directorios y comprobar que los permisos de los directorios son los adecuados.

- Base de datos: Recuperar la copia de seguridad de la base de datos, y restaurar los permisos correspondientes.

Realizando estos pasos, el sistema adquiere un extra de seguridad ante posibles contratiempos permite la recuperación del sistema a un punto determinado del tiempo. Dicho punto de restau-

ración permitirá realizar entornos similares de pruebas, así como recuperar el sistema en caso de desastre.

6.7. En este capítulo

Se ha llevado a cabo la instalación de todo el software necesario para la consecución del proyecto. Los pasos han sido detallados en su totalidad y cada pieza de software ha sido configurada para interactuar con el resto.

Se ha hecho especial hincapié en la seguridad, habilitando reglas específicas para el cortafuegos así como la utilización de protocolos de transmisión encriptados bajo ssl, para asegurar la confidencialidad e integridad de los mensajes.

Por último se integrarán una serie de pruebas para comprobar que todo el software funciona correctamente, se conectarán desde distintos dispositivos y programas software, que realizarán las actualizaciones pertinentes, para comprobar que la sincronización es correcta y se produce de forma satisfactoria.

7

Resultados

En este penúltimo capítulo se van a resumir la serie de pruebas que se llevarán a cabo contra el servidor, dichas pruebas comprenden desde test de seguridad, pasando por la funcionalidad y facilidad de gestión que se pretendía conseguir con el proyecto.

Con esta información se obtendrán conclusiones acerca del funcionamiento global del sistema, en caso de que alguno de los objetivos no fuese cumplido, se realizarán las oportunas modificaciones.

7.1. Comprobación del sistema

El sistema deberá cumplir una serie de requisitos para dar por finalizada su implantación. Es por ello que para cada una de las partes instaladas se definen una serie de tareas o reglas que se deben cumplir en su totalidad, antes de dar el visto bueno a la implementación del sistema. Para comprobar todas y cada una de la tareas, se realizan una serie de consideraciones que deberán cumplirse, para que el resto de pruebas puedan llevarse a cabo. Una vez cumplidas las consideraciones previas podrán ejecutarse el resto de verificaciones sobre la totalidad de servicios que se ofrecen.

7.1.1. Consideraciones previas

- Tras la instalación del sistema operativo (SO), se almacena de forma segura y correcta el usuario y la contraseña de la cuenta que tendrá permisos de administración sobre el sistema.

- Los usuarios administradores poseen los programas necesarios para acceder a las herramientas de administración de los servicios.
- Se dispone de una correcta instalación de red con todos los elementos necesarios, así como un correcto enrutamiento y direccionamiento.
- Se tienen definidas de forma concreta la lista de direcciones IP que podrán tener acceso de administración al sistema.
- En caso de disponer algún cortafuegos hardware, se definirán las reglas necesarias para una mayor protección del sistema.
- Todos los ficheros de log deberán almacenarse en la carpeta del sistema */var/log*.
- Existen los usuarios necesarios en el directorio Lightweight Directory Access Protocol (LDAP) con la información indispensable, como son el nombre de usuario, la contraseña y la dirección de correo electrónico.
- Se configuran toda clase de herramientas externas con los datos del servidor para su posible conexión o sincronización.

7.1.2. Securización del servidor

En primer lugar se comprueba si el sistema está totalmente securizado y cumple unos estándares de calidad. Es por ello que se realizan las siguientes comprobaciones sobre el mismo:

Estado	Regla	Anotaciones
OK	Sólo se puede entrar en la consola del sistema con los usuarios controlados mailuser y root, este último a través de *sudo*	El resto de usuarios de servicios no dispone de contraseña configurada, o no poseen consola de entrada
OK	La consola de administración por ssh sólo está accesible desde direcciones ip controladas	Están definidas las reglas de IPTables y securizado en el fichero */etc/host.allow* y */etc/host.deny*

Continúa ...

Estado	Regla	Anotaciones
OK	Sólo están abiertos los puertos para acceder a los servicios concretos de Hypertext Transfer Protocol (HTTP), Mail Transport Agent (MTA) y Mail Delivery Agent (MDA).	Las reglas de IPTables están definidas correctamente. El resto de servicios no podrá ser utilizado.
OK	Siempre se utilizan conexiones cifradas entre el servidor y el cliente para todas las comunicaciones.	En todas las comunicaciones se obliga a utilizar cifrado Secure Socket Layer (SSL) o Open Secure Shell (OpenSsh), bien redireccionando la petición, o rechazando las comunicaciones en claro.
OK	Las contraseñas no pueden ser almacenadas en claro.	Las contraseñas son guardadas en un fichero codificado cuya clave no consta en claro en ningún soporte físico.
OK	Cualquier intento de vulnerar el sistema deberá ser almacenado en una bitácora o log	En el directorio */var/log* se encuentran todos los ficheros donde se almacena dicha información. Además el cortafuegos está configurado para enviar al fichero de log los intentos de entrada no permitidos.

Tabla 7.1: Checklist para securización del servidor

7.1.3. Creación de certificados SSL

En segundo lugar, se comprobará que la creación de certificados se realiza de forma correcta, así como que se mantienen actualizados los valores de las bases de datos de certificados. Por último habrá que asegurar que los certificados son identificados de forma correcta por el software que va ser utilizado, y que la entidad certificadora puede exportarse e instalarse en distintos dispositivos.

Estado	Regla	Anotaciones
OK	Creación de nuevos certificados.	Se crean nuevos certificados de forma correcta, además las bases de datos se actualizan a la par.
OK	Instalación de la entidad certificadora en distintos dispositivos y/o software.	Se instala el certificado en SO como Ubuntu y Windows, además de en dispositivos Iphone y Android.
OK	Comprobar que los certificados que devuelve el servidor son reconocidos por la entidad certificadora instalada en el paso anterior.	Se utilizan las herramientas de SSL para comprobar dichos certificados de forma satisfactoria.

Tabla 7.2: Checklist para certificados SSL

7.1.4. Servicios secundarios

A continuación se enumerarán las pruebas a realizar sobre los servicios secundarios, dichos servicios comprenden el servidor HTTP, el Sistema de Gestión de Base de Datos (SGBD), el servidor LDAP y el servidor Domain Name System (DNS).

Estado	Regla	Anotaciones
OK	Comprobación del funcionamiento de los distintos servicios, arranque junto con el SO y correcto apagado.	Los servicios se arrancan correctamente junto al SO, no hay ninguna entrada de error en los logs del sistema.
OK	Comprobación que los servicios responden a las peticiones desde la máquina local.	Se comprueba que los puertos por los que escuchan los servicios están disponibles para peticiones locales, y responden de forma correcta.

Continúa ...

Estado	Regla	Anotaciones
OK	Comprobación que el servicio HTTP y DNS responden a peticiones externas de forma correcta.	Los servicios responden de forma correcta desde cualquier dirección a las solicitudes planteadas.
OK	Comprobación de logs de los servicios.	Todos los servicios dejan constancia de las tareas que realizan en los ficheros de logs de cada uno de ellos.
OK	Comprobación de redirección de HTTP de a Secure Hypertext Transfer Protocol (HTTPS).	El servidor web redirige todas las peticiones realizas por puerto no seguro, al seguro.
OK	Comprobación del correcto funcionamiento del SGBD.	El servicio funciona correctamente, se pueden realizar cambios en las bases de datos del programa, modificar los datos existentes, crear nuevos usuarios, asignar permisos, etc.
OK	Comprobación del correcto funcionamiento del servidor LDAP.	El servidor tan sólo es accesible de forma local. Las búsquedas y modificaciones del árbol de forma local se hacen correctamente.
OK	Comprobación de la herramienta para manejo del árbol LDAP.	La herramienta web es sólo accesible desde unas direcciones determinadas, y securizada mediante la solicitud de nombre de usuario y contraseña. Una vez logados, el usuario puede realizar las tareas correspondientes en el árbol. Se dan de alta/modifican/baja usuarios y grupos de forma satisfactoria.
OK	Comprobación del correcto funcionamiento del servidor DNS.	El servidor permite el alta/baja/modificación de las entradas de DNS configuradas.

Continúa ...

Estado	Regla	Anotaciones
OK	Comprobación que los usuarios pueden modificar sus datos personales.	El servidor permite a los usuarios modificar sus datos personales tras introducir su nombre de usuario y su contraseña, a través de la interfaz web.

Tabla 7.3: Checklist para servicios secundarios

7.1.5. Servicio de Correo

Para comprobar que el sistema groupware está funcionando de forma correcta, hay que testear de forma intensa uno de sus principales servicios, el servidor de correo. Dentro de la serie de procedimientos a comprobar en el servidor de correo, deberá incluirse la conectividad entre las distintas herramientas que componen dicho sistema, así como los requisitos de seguridad exigibles que deben cumplirse.

Todas aquellas peticiones en las que se requiera se utilizará un nombre de usuario con su contraseña asociada de forma correcta.

Estado	Regla	Anotaciones
OK	Comprobación del funcionamiento de los distintos servicios, arranque junto con el SO y correcto apagado.	Los servicios se arrancan correctamente junto al SO, no hay ninguna entrada de error en los logs del sistema.
OK	Comprobación que los servicios responden a las peticiones desde la máquina local.	Se comprueba que los puertos por los que escuchan los servicios están disponibles para peticiones locales, y responden de forma correcta.
OK	Comprobar si se produce la conexión entre las distintas partes del software y los servicios.	El software MTA y MDA se conectan correctamente entre sí. Además la autenticación es delegada de forma correcta al directorio LDAP.

Continúa ...

Estado	Regla	Anotaciones
OK	Comprobación que el servicio MTA y MDA responden a peticiones externas de forma correcta.	Los servicios responden de forma correcta desde cualquier dirección a las solicitudes planteadas.
OK	Comprobar si el software Mail User Agent (MUA) se conecta de forma correcta al servicio.	Con los datos correctos, el software se conecta al servicio.
OK	Se envían y se reciben correos electrónicos en el servidor.	El servidor es capaz de enviar y recibir correos desde cuentas autenticadas. También es capaz de enviar correos a cuentas externas.
OK	Comprobación de los sistemas de antivirus y antispam.	El software especificado trabaja de forma correcta, comprueba los correos tanto enviados como recibidos, y su funcionamiento es el adecuado.
OK	Comprobar los directorios locales de los usuarios.	Los directorios son creados de forma correcta y con los permisos específicos. Se almacenan los correos de la forma adecuada.

Tabla 7.4: Checklist para el servicio de correo.

7.1.6. Servicio Groupware

El servicio groupware es el principal reclamo del servidor. Dicho servicio permitirá de forma sencilla que los usuarios puedan comunicarse entre ellos eficazmente. Además la posibilidad de sincronización entre distintas herramientas hará de este servicio algo esencial en el día a día del intercambio de información entre los empleados. Es por ello que las distintas pruebas a las que debe ser sometido deben ser especialmente meticulosas, así como realizadas desde distintos dispositivos y/o usuarios.

Estado	Regla	Anotaciones
OK	Comprobar que el sistema está funcionando correctamente.	El sistema está levantado sobre el servidor HTTP de forma correcta.
OK	Acceso al sistema y autenticación de los usuarios.	El sistema ofrece la pantalla de autenticación y se accede a la zona del usuario tras poner los datos correspondientes.
OK	Conexión con el resto de sistemas.	El sistema se comunica con el resto de servicios y ofrece la información correcta en cada momento.
OK	Creación de los datos del sistema.	El sistema accede de forma correcta al SGBD y tiene los permisos necesarios para su correcto funcionamiento.
OK	Sincronización entre dispositivos a través de los protocolos *exchange*.	El sistema ofrece sincronización completa entre los distintos dispositivos, además que realiza todas las actualizaciones de forma correcta, tanto de contactos como de agenda.
OK	Modificación de los datos del servidor para cada uno de los usuarios.	El sistema ofrece distintas forma de acceso a los datos, además de permitir su actualización desde distintas fuentes, teniendo todos los dispositivos de acceso sincronizados.
OK	Creación de cuentas de administración.	El sistema ofrece la posibilidad de la asociación de cuentas ya creadas con los roles de administración.
OK	Creación de usuarios.	El usuario podrá conectarse al sistema siempre que posea una cuenta en el servidor de directorio LDAP. Los datos asociados a dicha cuenta serán creados automáticamente.

Tabla 7.5: Checklist para el servicio groupware.

7.1.7. Análisis de los resultados

Los resultados de todas las pruebas han concluido de forma satisfactoria. De esta forma se puede asegurar que el sistema cumple con todos y cada uno de los requisitos planteados en la sección Necesidades en la página 90. De esta forma y a modo de resumen el sistema ofrece:

- **Servicio de correo electrónico**, el sistema permite el envío y la recepción de mensajes de correo entre el personal de la empresa, así como la comunicación con otras direcciones externas. El sistema es compatible con los protocolos Simple Mail Transfer Protocol (SMTP), Internet Message Access Protocol (IMAP) y Post Office Protocol (POP). El servidor es compatible con los protocolos cifrados de los mismos. El servidor debe incluye entre sus características el filtrado de correo, así como la detección de correo no deseado. Dicho servidor es altamente configurable además de estar basado en protocolos estándar para su compatibilidad con la mayoría de los clientes, y de cara a posibles exportaciones o importaciones.

- **Identificación centralizada**, los usuarios se identifican en sus cuentas utilizando las mismas credenciales utilizadas para el resto de servicios ofrecidos por la empresa.

- **Servicio de agenda y directorio personal**, el sistema cuenta con una base de datos donde el usuario podrá almacenar y compartir la información referente a su agenda personal, así como un directorio donde almacenar datos de sus contactos. Dichos datos pueden ser accedidos desde cualquier dispositivo conectado a internet.

- **Servicio DNS**, el sistema da soporte para resolver los nombres y direcciones IP de todos los servicios que alberga, así como nombres relativos al resto de la empresa. También posee de un servicio de caché de DNS para centralizar todas las peticiones y así minimizar el tráfico DNS hacia internet.

- **Entidad certificadora**, existe una entidad certificadora para la creación y verificación de certificados capaces de cifrar las conexiones de los servicios mediante SSL. El certificado de dicha entidad podrá ser instalado en todos los equipos que requieran utilizar los servicios por protocolo cifrado, para evitar problemas de confianza entre el software cliente y el servidor.

- **Servicio WEB**, el servicio web da acceso a las herramientas de correo web para los usuarios, así como ofrece la posibilidad de la instalación de herramientas de administración vía web para el software del servidor.

- **Servicio de base de datos**, todos los datos son almacenados y mantenidos en un SGBD. Dicho gestor está disponible para los distintos servicios, así como para almacenar información de las herramientas de gestión, incluso para almacenar cierta información de los usuarios.

Además, siguiendo las orientaciones marcadas en dicha sección también se puede llegar a la conclusión de los siguientes puntos:

- **Comunicaciones unificadas**, el entorno groupware provee de las herramientas necesarias para la comunicación desde todos los miembros de la empresa. Dicha comunicación es independiente de las herramientas utilizadas así como de los dispositivos desde las que se acceden.

- **Alta disponibilidad**, gracias a la virtualización, el sistema permanece disponible el máximo tiempo posible.

- **Administración sencilla**, el sistema posee distintas herramientas de administración, para distintos niveles de administrador. Entre dichas herramientas se pueden encontrar herramientas web que ofrecerán una interfaz sencilla para los administradores más inexpertos.

- **Conjunto de servicios**, el conjunto de servicios que ofrece el sistema se puede resumir en:

  - Servidor de correo electrónico con soporte para los protocolos SMTP, POP y IMAP.
  - Servidor de páginas web, para albergar las herramientas de administración, tanto para usuarios como administradores.
  - Servidor de agenda y directorio, para almacenar la información personal de los usuarios.

- **Seguridad y confidencialidad**, el sistema es accesible mediante protocolos que usan encriptación para una mayor seguridad y confidencialidad de los datos. Además se obliga siempre a su utilización por parte del usuario.

7.2. En este capítulo

En este capítulo se han planteado las distintas pruebas que debía pasar el servidor para corroborar su perfecto funcionamiento. Dichas pruebas deberán ser realizadas a todos aquellos servidores en los que se instale el conjunto de todas las herramientas. Una vez comprobados todos los puntos, y sobrepasados los mismos, se concluye que el servidor cumple con todas y cada una de las especificaciones marcadas al inicio del proyecto, por lo tanto se remarca la satisfacción de haber realizado un buen trabajo.

8

Conclusiones y trabajos futuros

Por último, en este capítulo, se profundizará sobre las conclusiones finales sobre el proyecto, lo que se ha conseguido y lo que se ha tenido que descartar, así como una valoración personal de la realización del mismo. Además se incluyen posibles trabajos futuros a llevar a cabo una vez que el sistema se encuentre en funcionamiento, y que dependerá en gran medida de la capacidad de crecimiento del sistema, así como la utilización del mismo.

8.1. Conclusiones

Llegados a este punto toca realizar un resumen de los trabajos que se han llevado a cabo, así como las conclusiones sacadas de la realización del mismo.

Para llevar a cabo un proyecto de este alcance se han tenido muy en cuenta los siguientes puntos:

- En la realización del proyecto fin de carrear se han incluido referencias a fuentes solventes y reputadas en el ámbito del mismo, que están disponibles en la Bibliografía.
- Se ha definido el problema y se ha ofrecido la base de conocimiento necesaria para su entendimiento en el Estado de la cuestión.
- El Presupuesto es ajustado, contemplando los aspectos necesarios para realizar una estimación real de los costes de realización del mismo.

El proyecto se ha podido llevar a cabo gracias a la mayoría de las habilidades o conocimientos básicos adquiridos durante los estudios universitarios cursados, y concretamente los conocimientos de:

- **Servidores de Información**: Con la documentación aportada por la asignatura se sientan las bases de la comunicación entre todos los servicios que componen este proyecto.
- **Ingeniería del Software**: Dicha asignatura ha sido fundamental para la captación de requisitos, así como para definir el plan de pruebas a realizar sobre el sistema.
- **Sistemas Informáticos**: Toda esta documentación ha sido basada en las prácticas realizadas en esta asignatura.
- **Seguridad en las Tecnología de la Información**: La seguridad así como el cumplimiento de la legalidad era un punto muy a tener en cuenta, y cuya bases fueron sentadas por esta asignatura.
- **Seguridad en los Sistemas Distribuidos**: Donde se hacía hincapié en la necesidad de de utilizar protocolos encriptados para la transmisión de la información.
- **Redes de Ordenadores**: Gracias al conocimiento de los distintos protocolos así como el funcionamiento de las comunicaciones explicado en esta asignatura, ha sido posible definir de forma efectiva todas las comunicaciones entre las herramientas, así como definir las reglas necesarias a nivel de cortafuegos.
- **Diseño de Sistemas Operativos**: Esencial para llevar a cabo la realización de cualquier proyecto referido a la informática. Los conocimientos sobre el propio funcionamiento de cualquier sistema operativo (SO) es totalmente necesario a la hora de implantar cualquier solución.
- **Administración de Empresas**: Dicha asignatura ayuda a conocer el entramado empresarial así como conocer la realización de presupuestos y estimaciones para conseguir el mayor rendimiento de los recursos disponibles.

Por otra parte, gracias a la amplia experiencia recogida en las distintas Becas ofrecidas por los Departamentos de la Universidad Carlos III, y en concreto en el Laboratorio del Departamento de

Informática, este trabajo ha podido realizarse de una forma más efectiva y sobre unas bases muy sólidas en la administración de servicios, usuarios y SO.

Por último la experiencia profesional adquirida en mi actual empresa y en los distintos trabajos realizados en la misma, le dan un toque más profesional si cabe al trabajo realizado, y la satisfacción del reconocimiento de que dicho trabajo no se va a quedar en un mero documento con unas pruebas, si no que va a perdurar durante un tiempo.

8.2. Opinión personal

El siguiente proyecto expone de forma muy concreta la importancia de las comunicaciones en el día a día. Es esencial que dichas comunicaciones puedan realizarse de forma segura, ofreciendo en todo momento la más estricta confidencialidad en la transmisión de los mensajes, y todo ello, sin comprometer la experiencia del usuario. Antes de la definición de este proyecto se planteó la cuestión de excoger algo que ya ofreciese el mercado, pero no existiendo ninguna solución que se acoplase cien por cien a lo exigido, se dispuso a la consecución del siguiente proyecto. Por lo tanto, no existe ninguna solución igual a la planteada, siendo esa una aportación original propia.

Durante la instalación y la búsqueda de información, el proyecto me ha enriquecido personalmente en los siguientes puntos:

- Conocimiento de administración de servidores. Ahondando en temas de seguridad y confidencialidad, pudiendo ofrecer al usuario final un completo entorno final seguro y confiable.
- Conocimiento de los protocolos de comunicación entre personas más importantes, como son los de correo. Así como la realización de conectividad entre distintos dispositivos para una completa sincronización.
- Me ha dado una visión personal a que hay vida más allá de las herramientas que ofrecen las grandes corporaciones como son Microsoft, Google o Apple, y que utilizando software libre se pueden conseguir productos con similares características, y las misma utilidad.
- Creación y administración de árboles de directorio, ahondado más si cabe en la utilización de los mismos para muchas aplicaciones, como puede ser autenticación, directorio telefónico,

utilización de certificados, etc.

- Afiancé conocimientos sobre técnicas de copias de seguridad, así como la utilización de distintas herramientas referentes a la administración de LDAP, MySQL, etc. Muy importantes para demostrar tu valía en el mundo profesional.

Además de todo lo comentado, la satisfacción de crear un proyecto sobre el que basar las comunicaciones actuales de la empresa, es uno de los mayores reconocimientos al trabajo realizado. De esta forma se reconoce que el trabajo es utilizado diariamente y el servidor cumple su cometido final, que es el de ayudar a las personas a estar comunicadas de una forma fácil y efectiva.

Por otra parte, el llevar el trabajo sobre un entorno empresarial privado, te hace aprender a trabajar con una presión muy diferente al mero hecho de tener que realizar tu trabajo para aprobar. Tú trabajo debe perdurar en el tiempo, y debe satisfacer las necesidades ahora y durante un determinado tiempo de amortización. De esta forma vas viendo la evolución del servidor, como se realizan las copias de seguridad de forma correcta, los problemas que los usuarios pueden tener con sus cuentas, como crear, o modificar cuentas, etc. Trabajos en los que realmente se ve la diferencia frente a un entorno académico, donde todo termina cuando el profesor califica el trabajo realizado.

Todo el trabajo llevado a cabo para realizar esta documentación ha sido realizado con LaTeX, lo cual ha sido una experiencia muy enriquecedora y de la que estoy muy satisfecho. El acabado del documento representa el esfuerzo llevado a cabo para aprender a utilizar la herramienta y conseguir que pequeños cambios lleguen a representar una excelente presentación.

Y por último comentar que el siguiente proyecto pueda servir como referencia a futuros alumnos para acumular conocimiento, así cómo para cualquier persona dispuesta a ponerse a cacharrear con una máquina virtual y un entorno propio para la creación de su propio servidor groupware, que tan sólo requiere algunos conocimientos de administración Linux, un poco de paciencia y ganas de trabajar.

8.3. Trabajos futuro

Por último y para conocer como seguir explotando un poco más si cabe el sistema implantado, se van a ofrecer distintas ideas de cómo extender la instalación aquí realizada:

- Creación de tareas de copias de seguridad más eficientes, utilizando herramientas conocidas como *bacula* que ofrece una integración completa con Sistema de Gestión de Base de Datos (SGBD), servidores de correos, archivos de configuración, etc.

- Creación de infraestructuras utilizando software automatizado como *puppet*. Dicha herramienta permite la creación de infraestructura conectada y configurada sin necesidad de realizar intervenciones manuales. Un gran avance para este proyecto podría ser la creación de los ficheros de configuración de *puppet* para desplegar todos los servicios comentados y realizar únicamente una pequeña personalización de los mismos para tener el sistema completo.

- Estudio de escalabilidad de los servidores, según la arquitectura estudiada es posible separar cada uno de los servicios en distintos servidores, es decir llevar a cabo un crecimiento vertical, pero sería muy interesante abarcar una arquitectura horizontal en algunos de los servicios.

- Instalación y utilización de más servicios interesantes para una infraestructura de este tipo, como podrían ser servidores de ficheros de log, la utilización de distintos proxys tanto para el correo como el contenido web para liberar de carga a los servidores finales.

- Monitorización de los servicios y los servidores. Por políticas de empresa se instala un servicio de monitorización básico que no ha sido incluido en el manual. Incluir una política de monitorización así como llevar a cabo la creación de un manual para casos de emergencia sería deseable.

8.4. Para finalizar

En este capítulo se recogen las conclusiones, opiniones y trabajos que podrían extender el proyecto. A sido un largo recorrido desde que comencé los estudios universitarios allá por el año 2000 y estas puede que sean mis últimas líneas para un documento académico. Espero que esto sólo sea un hasta luego y no un adiós definitivo.

El lector habrá encontrado un documento lleno de referencias, y con un detalle de los pasos que se han llevado a cabo hasta dar por finalizado el proyecto. En este último capítulo se condensa

la experiencia tanto académica como personal adquirida durante la realización del mismo, pero se puede dar por sentado que tan sólo es un pequeño grano de arena en la gran playa del conocimiento.

APÉNDICES

Siglas

AD Active Directory. 53, 54

ASP Active Server Pages. 57

CA Certification Authority. 28, 64, 65, 103, 154

CSR Certificate Signing Request. 64

DHCP Dynamic Host Configuration Protocol. 54

DKIM DomainKeys Identified Mail. 76

DMZ Demilitarized Zone. 69

DNS Domain Name System. 51, 54, 62–64, 76, 77, 87, 89, 94, 103, 113, 146–148, 186, 187, 191

FTP File Transfer Protocol. 57, 86–88

HTML HyperText Markup Language. 55, 66

HTTP Hypertext Transfer Protocol. 55–58, 86, 87, 127, 171, 178, 185–187, 190

HTTPS Secure Hypertext Transfer Protocol. 57, 100, 187

IIS Internet Information Services. 57

IMAP Internet Message Access Protocol. 41, 46, 47, 57, 89, 90, 92, 93, 100, 150, 191, 192

LDAP Lightweight Directory Access Protocol. 50–53, 93, 103, 141–143, 146, 151, 152, 157, 160, 164, 166–168, 175, 176, 178, 184, 186–188, 190

MDA Mail Delivery Agent. 41–44, 185, 188, 189

MTA Mail Transport Agent. 41, 46, 76, 92, 150, 151, 155, 185, 188, 189

MUA Mail User Agent. 41, 42, 189

NIS Network Information Service. 50

OpenSsh Open Secure Shell. 98, 117, 121, 124, 127, 185

PAM Pluggable Authentication Modules. 152

pear PHP Extension and Application Repository. 173

pecl PHP Extension Community Library. 173

POP Post Office Protocol. 41, 46, 47, 57, 89, 90, 92, 93, 100, 150, 191, 192

PWS Personal Web Server. 57

RDP Remote Desktop Protocol. 80

SGBD Sistema de Gestión de Base de Datos. 58, 59, 61, 90, 91, 94, 103, 138, 139, 186, 187, 190, 192, 198

SMTP Simple Mail Transfer Protocol. 40, 41, 43, 44, 57, 89, 90, 92, 100, 150, 157, 191, 192

SO sistema operativo. 31–35, 37–39, 42, 75, 78, 80, 81, 87, 88, 91, 92, 97, 103, 111, 113, 115, 117, 120, 138, 164, 183, 186, 188, 196

SSL Secure Socket Layer. 65, 89, 100, 127, 136, 146, 154, 157, 168, 171, 173, 185, 186, 191

TLS Transport Layer Security. 65

URI Uniform Resource Identifier. 77

Glosario

cortafuegos Un corta fuegos o firewall es una parte de un sistema o una red que está diseñada para bloquear el acceso no autorizado, permitiendo al mismo tiempo comunicaciones autorizadas. 68–70, 88, 98, 102, 146, 149, 169, 181, 184, 185, 196

exchange Protocolo utilizado por el servicio del mismo nombre y propiedad de Microsoft. Dicho protocolo permite el intercambio de mensajes de correo electrónico, sincronización de agenda, entre otros. 178

framework Conjunto de software que suele incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto. 47, 67, 68, 93

groupware El término groupware hace referencia a los métodos y herramientas de software que facilitan el trabajo en grupo, mejorando su rendimiento, y contribuyen a que personas que están localizadas en puntos geográficos diferentes puedan trabajar a la vez, ya sea directamente o de forma anónima, a través de las redes. 44, 47, 90, 100, 102, 170, 174–176, 188, 189, 192, 198

hipervisor Un hipervisor monitor de máquina virtual es una plataforma que permite aplicar diversas técnicas de control de virtualización para utilizar, al mismo tiempo, diferentes sistemas operativos en un mismo ordenador.. 78, 80, 81, 97

IPTables Conjunto de tablas y reglas del firewall a nivel de kernel implentado en Linux. Permite el filtrado de los paquetes de red. 69–71, 123, 125, 126, 184, 185

Long Time Support Versiones cuyo soporte se extiende más allá de la vida normal de las actualizaciones que soporta Canonical. Usualemtente hasta 5 años. 92, 97, 111

Logical Volume Manager Administrador logico de volúmenes de discos para el kernel de Linux. 97, 115

proxy Un servidor proxy es un ordenador que intercepta las conexiones de red que un cliente hace a un servidor de destino.. 57, 86, 102

TCP Wrapper Envoltorio TCP usado para filtrar el acceso a red de servicios. 123

Ubuntu Distribución Linux basada en Debian y mantenida por la empresa Canonical. 92, 97, 111, 127, 138, 139

user friendly Entorno o herramienta creada por humanos capaz de ofrecer al usuario la facilidad necesaria para alcanzar un objetivo concreto. 34, 43, 57

Bibliografía

[1] Ismael Callejas. *Convierte tu viejo Android en un servidor multiuso con Servers Ultimate*. Feb. de 2013. URL: http://www.elandroidelibre.com/2012/08/convierte-tu-viejo-android-en-un-servidor-multiuso-con-servers-ultimate.html.

[2] Tom Eastep. *Netfilter packet flow*. Feb. de 2013. URL: http://www.shorewall.net/images/Netfilter.png.

[3] Christoph Galuschka. *HowTos/Network/IPTables*. Jun. de 2012. URL: http://wiki.centos.org/HowTos/Network/IPTables.

[4] David Martínez Martí. *IPTables, qué es y como funciona*. Feb. de 2013. URL: http://www.sedice.com/modules.php?name=Forums&file=viewtopic&t=11496.

[5] Microsoft. *A history of Windows*. Ene. de 2013. URL: http://windows.microsoft.com/en-US/windows/history.

[6] Microsoft. *Ciclo de vida de soporte de Microsoft*. Ene. de 2013. URL: http://support.microsoft.com/lifecycle/default.aspx?LN=es-bo&C2=1163.

[7] Microsoft. *Microsoft Exchange Server 2010*. Feb. de 2013. URL: http://www.microsoft.com/spain/exchange/2010/overview.mspx.

[8] Microsoft. *Microsoft Windows Server 2008 Editions*. Ene. de 2013. URL: http://www.microsoft.com/es-es/server-cloud/windows-server/2008-r2-editions.aspx.

[9] Microsoft. *What Is Server Core?* Ene. de 2013. URL: http://technet.microsoft.com/en-us/library/dd184075.aspx.

[10] Microsoft. *Windows Server 2012 How to Buy*. Ene. de 2013. URL: http://www.microsoft.com/en-us/server-cloud/windows-server/buy.aspx.

[11] Vicente Navarro. *Crear los certificados SSL para nuestro servidor web HTTPS con Apache, OpenSSL y Debian Lenny*. Ene. de 2013. URL: http://www.vicente-navarro.com/blog/2009/02/22/crear-los-certificados-ssl-para-nuestro-servidor-web-https-con-apache-openssl-y-debian-lenny/.

[12] Marat Borisovich Nepomnyashy. *Using Git to Keep Track of Updates to Configuration Files in '/etc/'*. Mar. de 2013. URL: http://www.maratbn.com/blogs/2012/12/04/using-git-to-keep-track-of-updates-in-etc/.

[13] Netcraft. *Web Server Survey*. Ene. de 2013. URL: http://news.netcraft.com/archives/category/web-server-survey/.

[14] Rusty Russell. *IptablesHowTo*. Jun. de 2012. URL: https://help.ubuntu.com/community/IptablesHowTo.

[15] John Thompson. *How to install Android 2.3 on the Raspberry Pi*. Feb. de 2013. URL: http://reviews.cnet.co.uk/desktops/how-to-install-android-2-3-on-the-raspberry-pi-50009931/.

[16] Carnegie Mellon University. *Project Cyrus*. Feb. de 2013. URL: http://cyrusimap.web.cmu.edu/.

[17] Dovecot Wiki. *Authentication Mechanisms*. Feb. de 2013. URL: http://wiki2.dovecot.org/Authentication/Mechanisms.

[18] Wikipedia. *La catedral y el bazar*. Mar. de 2013. URL: http://es.wikipedia.org/wiki/La_catedral_y_el_bazar.

[19] Wikipedia. *LAMP (software bundle)*. Ene. de 2013. URL: http://en.wikipedia.org/wiki/LAMP_(software_bundle).

[20] Wikipedia. *LDAP*. Feb. de 2013. URL: http://es.wikipedia.org/wiki/LDAP.

[21] Wikipedia. *List of Linux distributions*. Ene. de 2013. URL: http://en.wikipedia.org/wiki/List_of_Linux_distributions.

[22] Wikipedia. *Logical Volume Manager (Linux)*. Dic. de 2012. URL: http://en.wikipedia.org/wiki/Logical_Volume_Manager_(Linux).

[23] Wikipedia. *Versiones Max OS X*. Feb. de 2013. URL: http://es.wikipedia.org/wiki/Mac_OS_X#Versioness.